Module 10: IoT Penetration Testing Methodology

Objective

The objective of this module is to help students learn how to assess IOT device firmware.

The following activities will be performed in the lab:

- Obtain a copy of the devices firmware
- Identify the type of firmware file system
- Explore the devices file system
- Analyze the types of files
- Run the firmware with an emulator

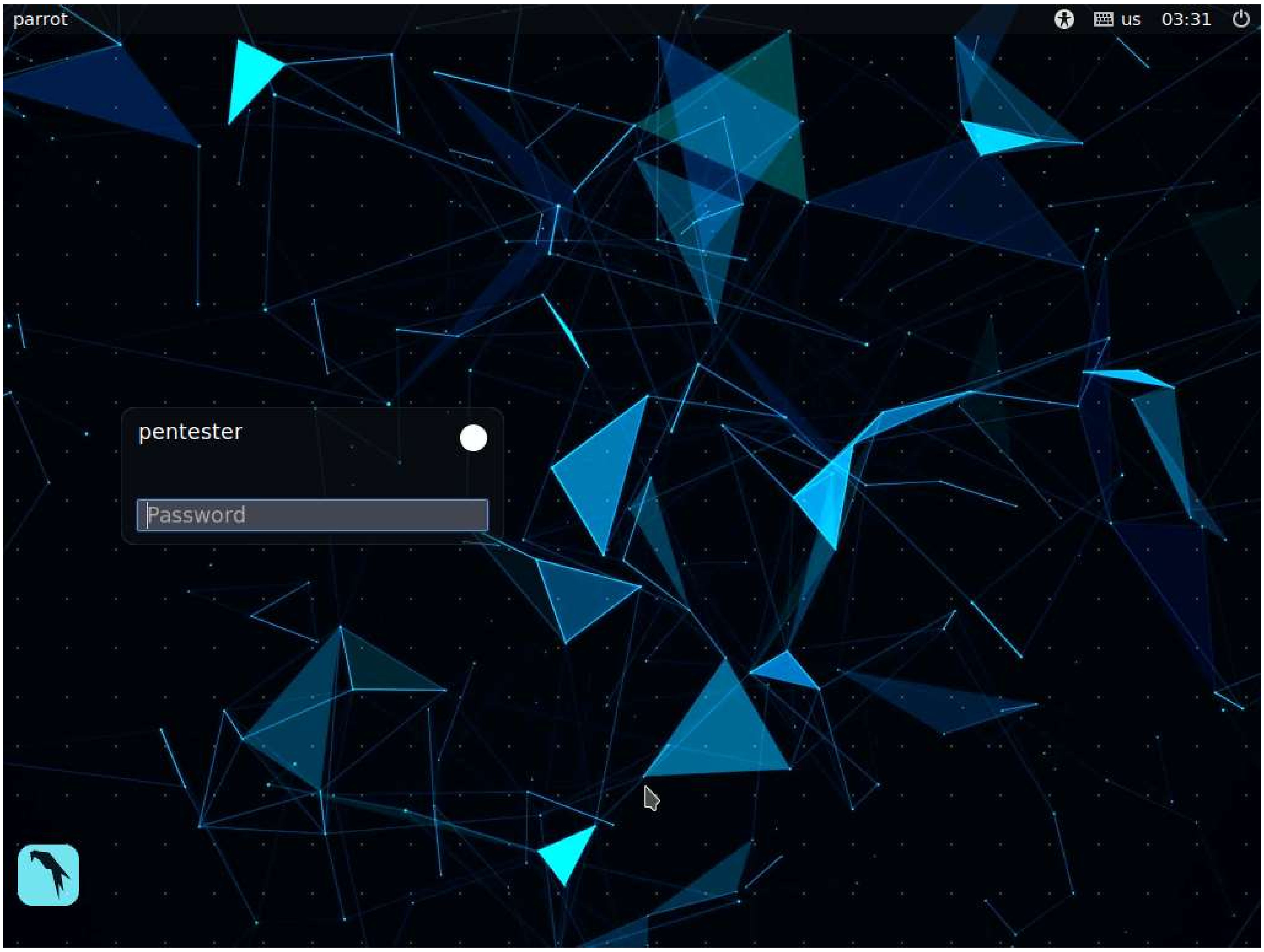
Scenario

This module is designed to provide the necessary practice and knowledge for locating and identifying vulnerabilities in IOT firmware. The ability to first acquire, and then extract and emulate an images firmware is a critical component for performing IOT hacking.

Exercise 1: IOT Firmware Acquisition, Extraction, Analysis and Emulation

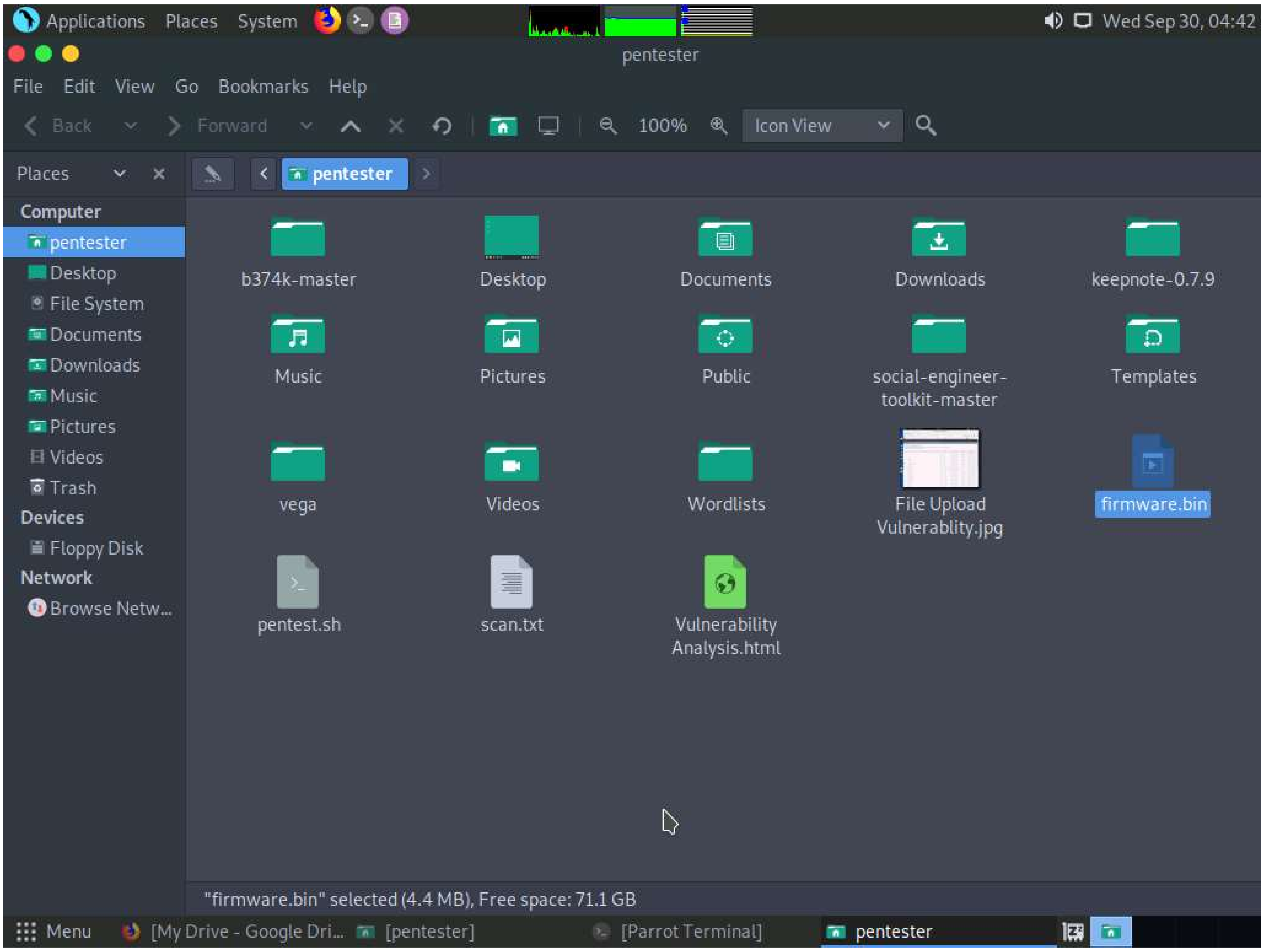
Lab Duration: **10** Minutes

1. By default **CPENT-M10 Parrot Security** machine appears. In the Password field type **toor** and press **Enter** to login.



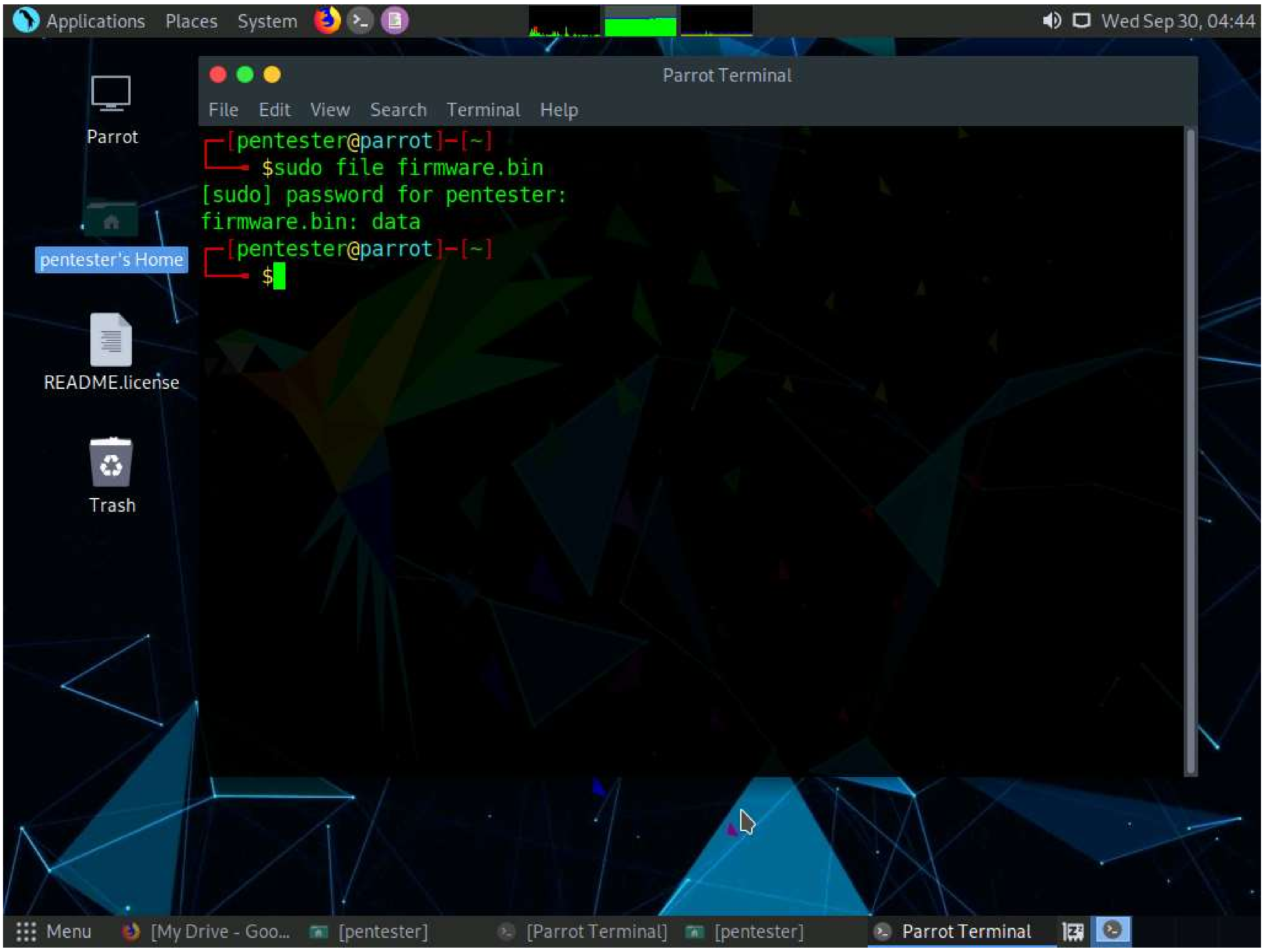
2. We have a sample firmware package binary for a **Netgear IOT** device that is often used for practicing IOT hacking. The device mo number is **WNAP 320**, which is a wireless access point for a network. An example of the device is shown in the following image.

3. We have downloaded the firmware for you. It is located in the **home** folder and named **firmware.bin**.



"firmware.bin" selected (4.4 MB), Free space: 71.1 GB

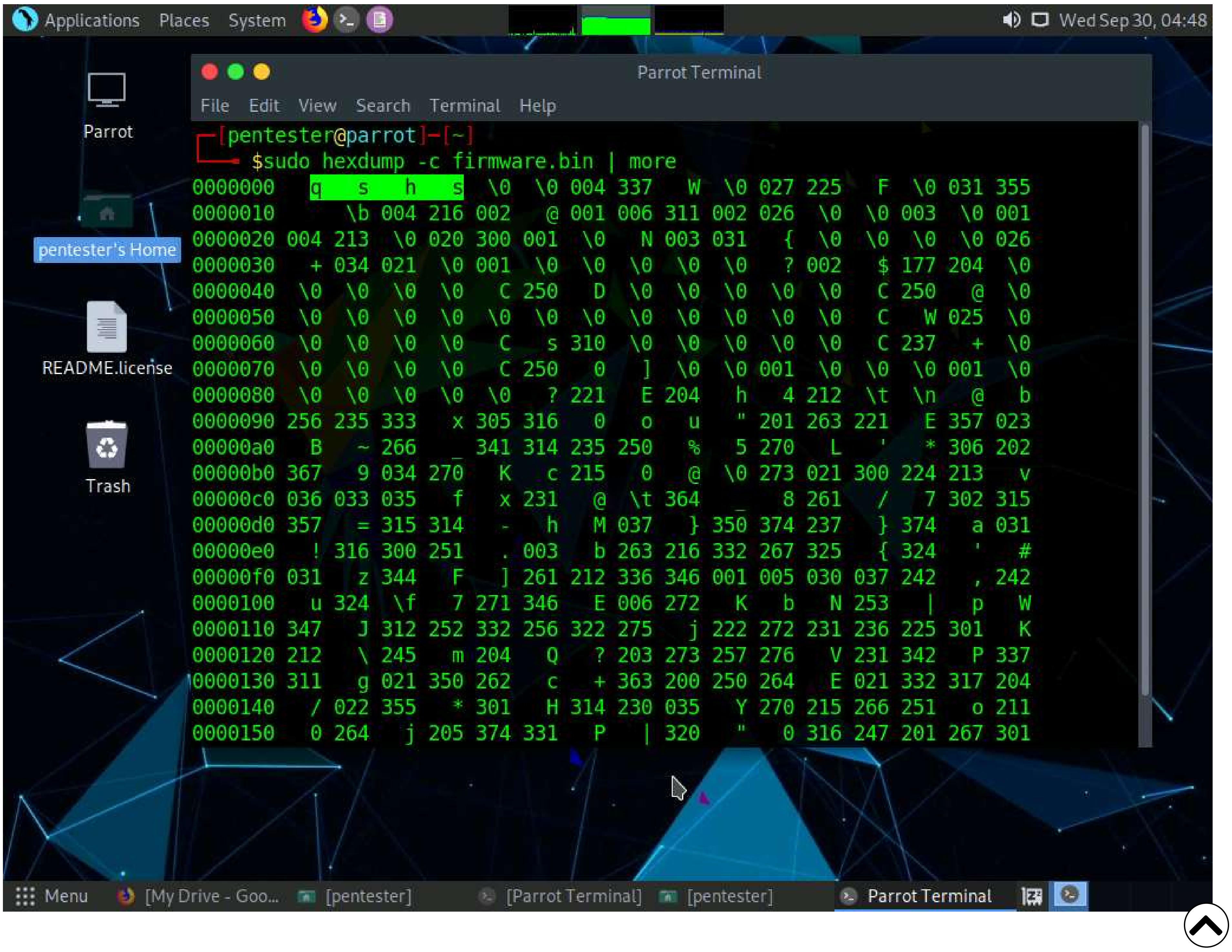
4. The first step when we have a file is to run the file command on it. Launch a Terminal and type **sudo file firmware.bin** and press **Enter**. Type **toor** when prompted for the password and press **Enter**.



```
┌─[pentester@parrot]─[~]
└──╼ $sudo file firmware.bin
[sudo] password for pentester:
firmware.bin: data
┌─[pentester@parrot]─[~]
└──╼ $
```

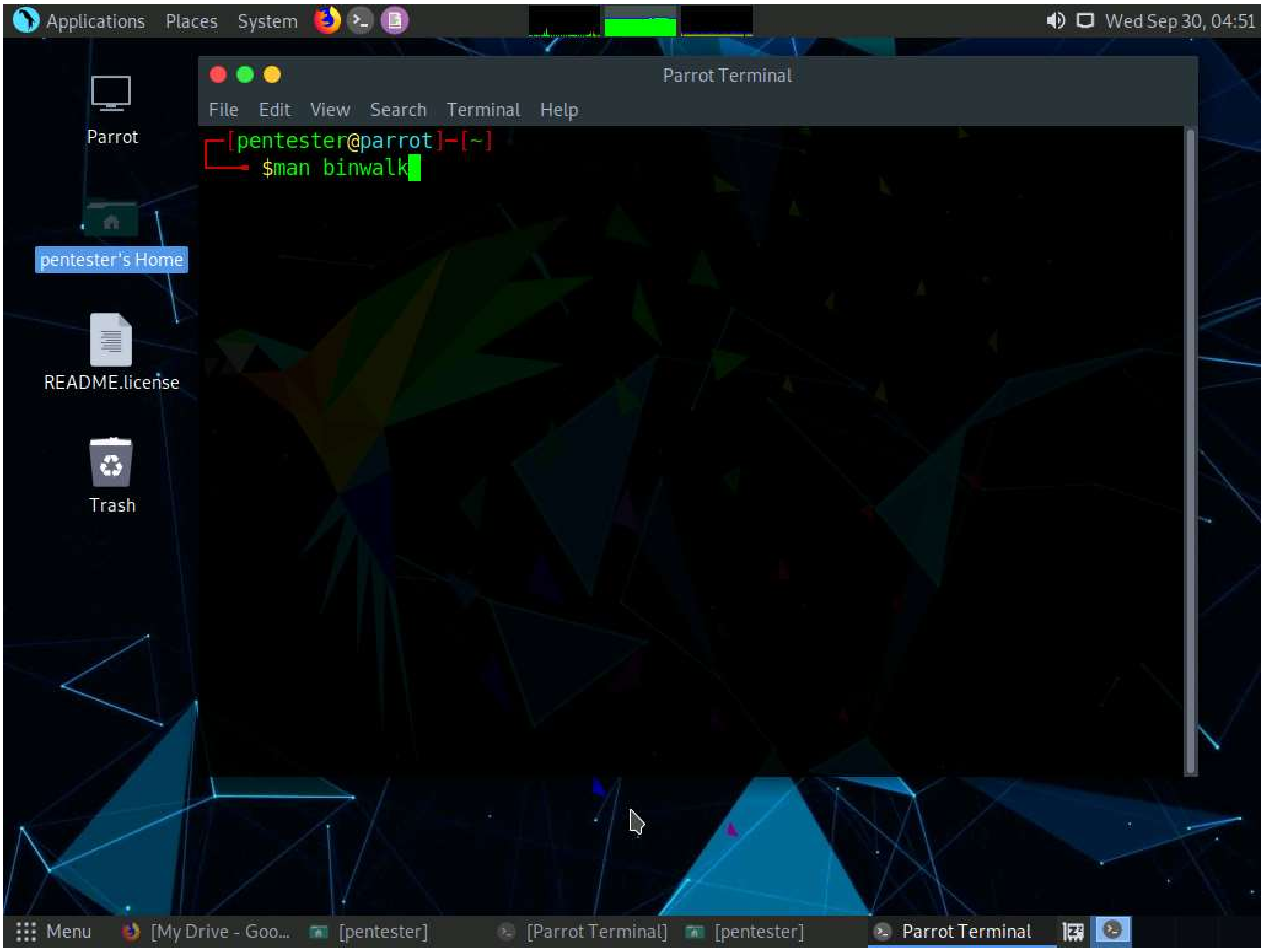
5. There is nothing useful for us here, so we continue trying. Type **sudo hexdump -c firmware.bin | more** and press **Enter**.

6. Now we have **qshs** that is identified in the hex trace. It is a known signature of the **Squashfs** file system, so this is a start.

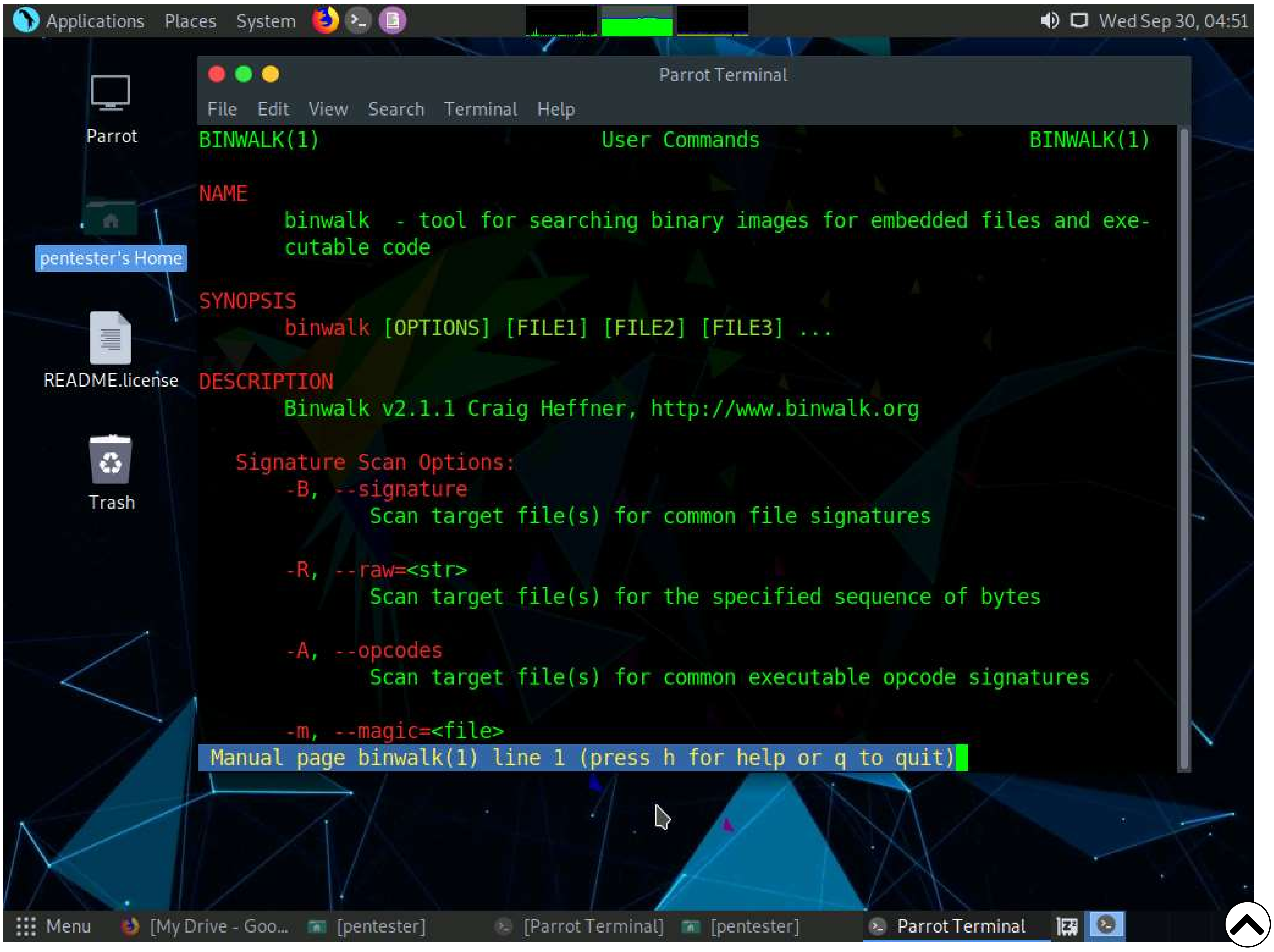


```
┌─[pentester@parrot]─[~]
└──╼ $sudo hexdump -c firmware.bin | more
0000000   q   s   h   s  \0  \0 004 337   W  \0 027 225   F  \0 031 355
0000010      \b 004 216 002       @ 001 006 311 002 026  \0  \0 003  \0 001
0000020 004 213  \0 020 300 001  \0   N 003 031   {  \0  \0  \0  \0 026
0000030   + 034 021  \0 001  \0  \0  \0  \0  \0   ? 002   $ 177 204  \0
0000040  \0  \0  \0  \0   C 250   D  \0  \0  \0  \0  \0   C 250   @  \0
0000050  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0   C   W 025  \0
0000060  \0  \0  \0  \0   C   s 310  \0  \0  \0  \0  \0   C 237   +  \0
0000070  \0  \0  \0  \0   C 250   0   ]  \0  \0 001  \0  \0  \0 001  \0
0000080  \0  \0  \0  \0  \0   ? 221   E 204   h   4 212  \t  \n   @   b
0000090 256 235 333   x 305 316   0   o   u   " 201 263 221   E 357 023
00000a0   B   ~ 266   _ 341 314 235 250   %   5 270   L   '   * 306 202
00000b0 367   9 034 270   K   c 215   0   @  \0 273 021 300 224 213   v
00000c0 036 033 035   f   x 231   @  \t 364   _   8 261   /   7 302 315
00000d0 357   = 315 314   -   h   M 037   } 350 374 237   } 374   a 031
00000e0   ! 316 300 251   . 003   b 263 216 332 267 325   { 324   '   #
00000f0 031   z 344   F   ] 261 212 336 346 001 005 030 037 242   , 242
0000100   u 324  \f   7 271 346   E 006 272   K   b   N 253   |   p   W
0000110 347   J 312 252 332 256 322 275   j 222 272 231 236 225 301   K
0000120 212   \ 245   m 204   Q   ? 203 273 257 276   V 231 342   P 337
0000130 311   g 021 350 262   c   + 363 200 250 264   E 021 332 317 204
0000140   / 022 355   * 301   H 314 230 035   Y 270 215 266 251   o 211
0000150   0 264   j 205 374 331   P   | 320   "   0 316 247 201 267 301
```

7. The next step in the process is to try and extract the file system from this firmware image. Our preferred tool is **binwalk**, so we will try this. In the terminal window, type **man binwalk** and press **Enter**. Take a few minutes to read about the tool. Once you are done, type **q** to exit the man page.

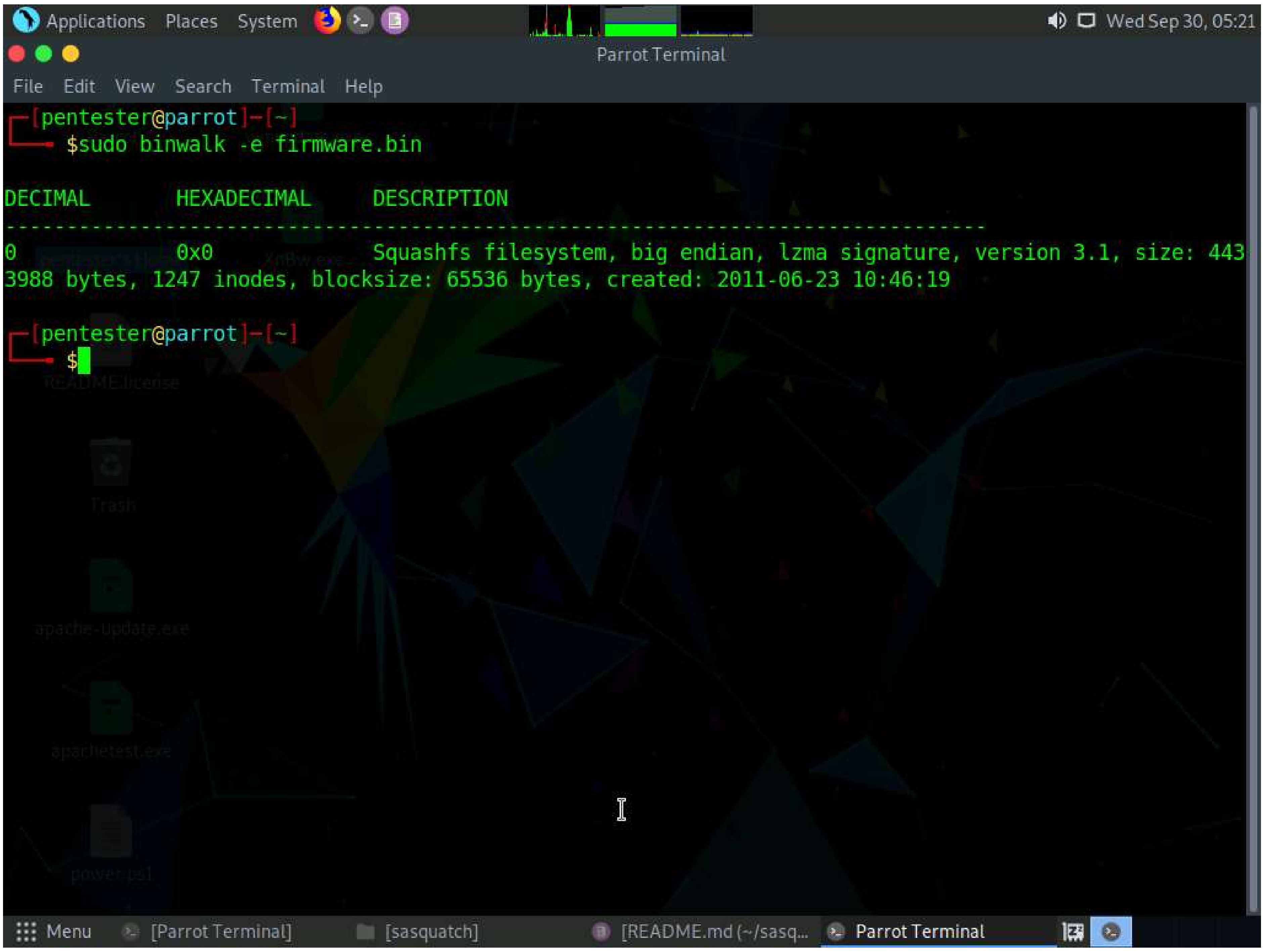


```
┌─[pentester@parrot]─[~]
└──╼ $man binwalk
```



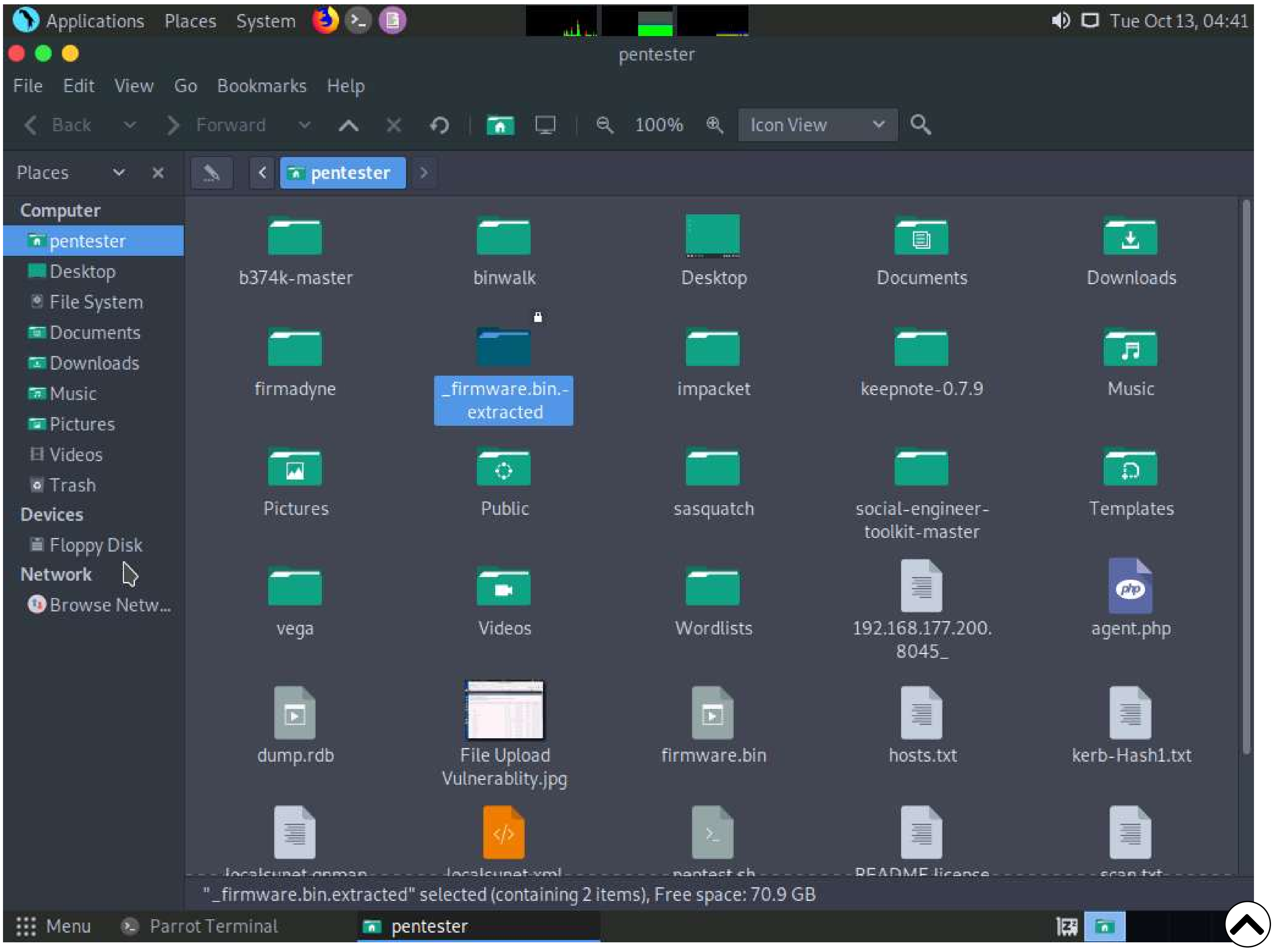
```
BINWALK(1)                    User Commands                    BINWALK(1)

NAME
       binwalk  - tool for searching binary images for embedded files and exe‐
       cutable code

SYNOPSIS
       binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

DESCRIPTION
       Binwalk v2.1.1 Craig Heffner, http://www.binwalk.org

   Signature Scan Options:
       -B, --signature
              Scan target file(s) for common file signatures

       -R, --raw=<str>
              Scan target file(s) for the specified sequence of bytes

       -A, --opcodes
              Scan target file(s) for common executable opcode signatures

       -m, --magic=<file>
 Manual page binwalk(1) line 1 (press h for help or q to quit)
```

8. In the terminal type **sudo binwalk -e firmware.bin** and press **Enter**. However, the main thing is the message that the image has been extracted. We also see that the filesystem is confirmed as Squashfs. Once the command is completed, you will have a directory that represents the extracted filesystem in the **Home** folder.



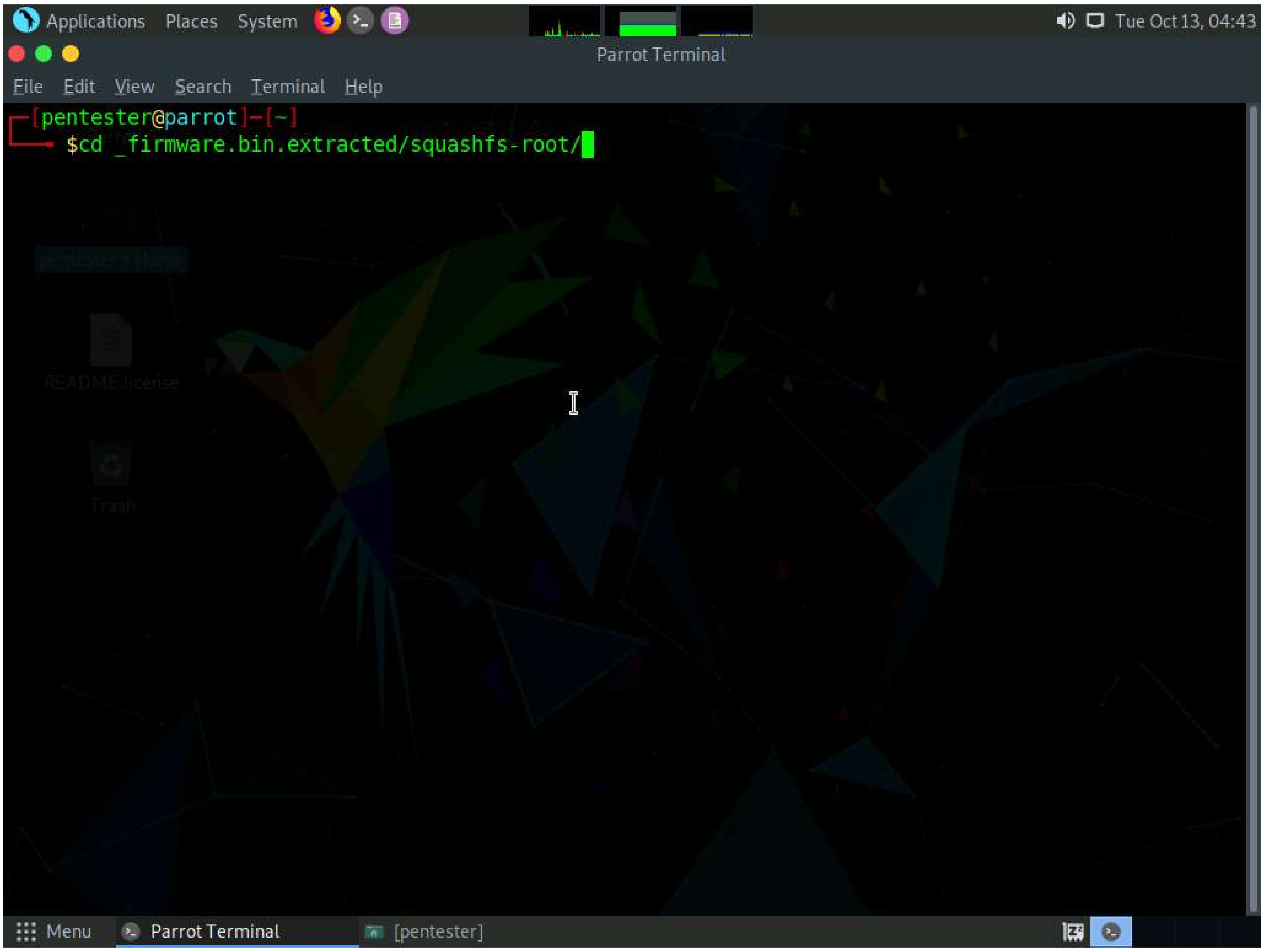
```
┌─[pentester@parrot]─[~]
└──╼ $sudo binwalk -e firmware.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Squashfs filesystem, big endian, lzma signature, version 3.1, size: 443
3988 bytes, 1247 inodes, blocksize: 65536 bytes, created: 2011-06-23 10:46:19

┌─[pentester@parrot]─[~]
└──╼ $
```

Applications Places System
Wed Sep 30, 05:21
Parrot Terminal
File Edit View Search Terminal Help
Menu [Parrot Terminal] [sasquatch] [README.md (~/sasq... Parrot Terminal

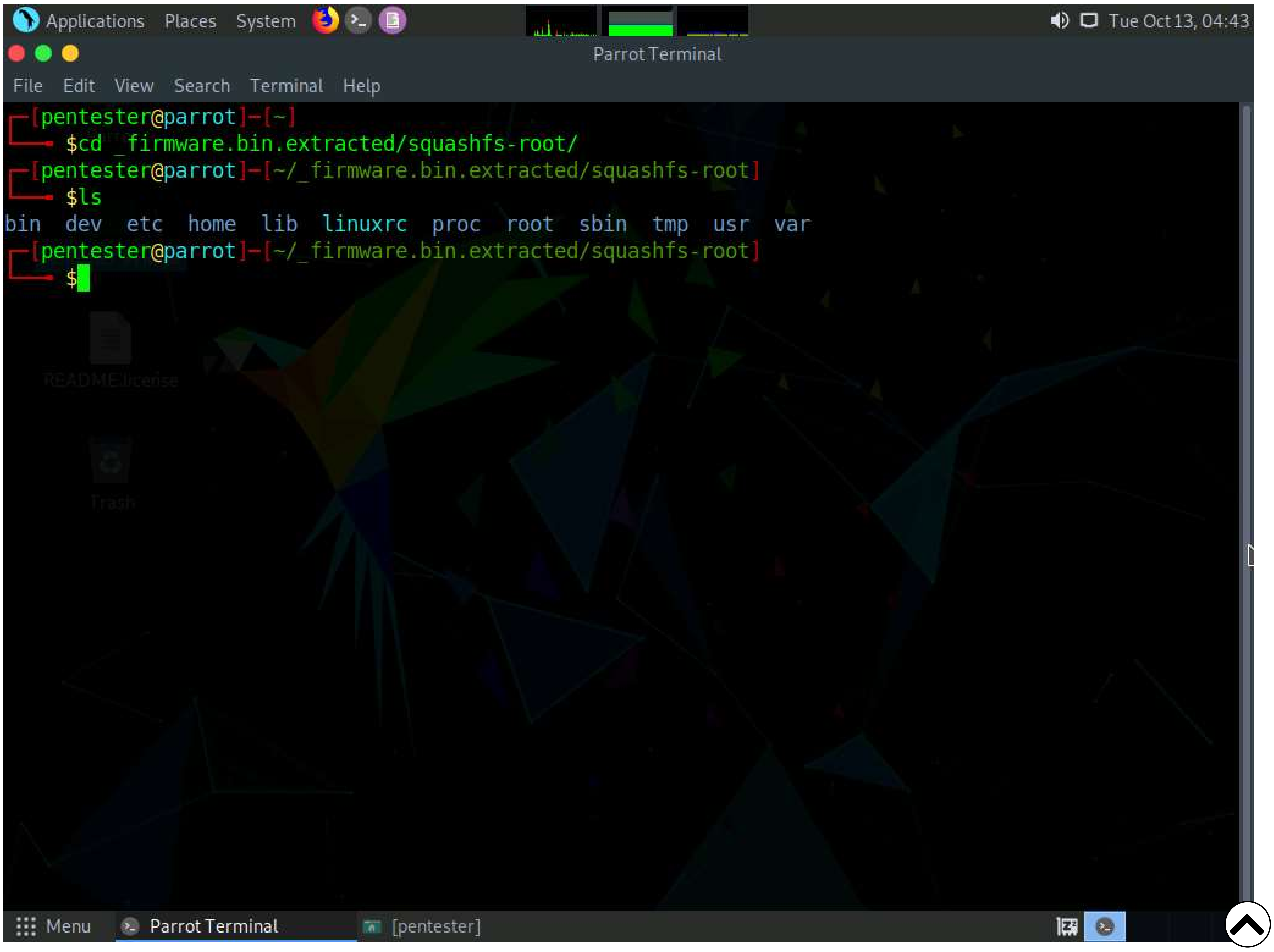


"_firmware.bin.extracted" selected (containing 2 items), Free space: 70.9 GB

9. In the terminal type **cd _firmware.bin.extracted/squashfs-root** and press **Enter**. Once you are in the directory type **ls** and press **Enter**. You will see that the extracted file system is in a squashfs-root directory. We now have a complete and extracted file system, which as you can see, is a Linux variant.

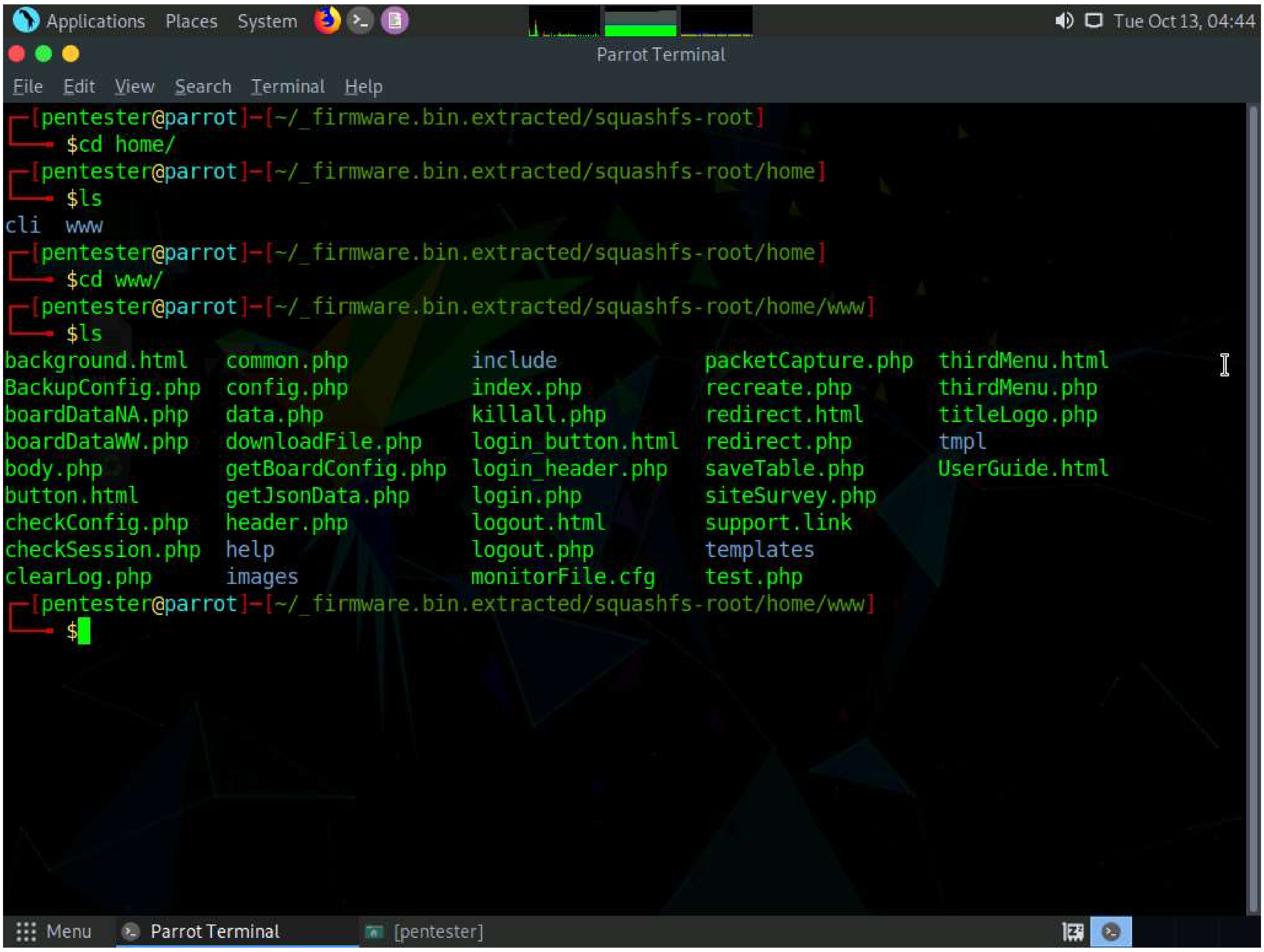


```
┌─[pentester@parrot]─[~]
└──╼ $cd _firmware.bin.extracted/squashfs-root/
```



```
┌─[pentester@parrot]─[~]
└──╼ $cd _firmware.bin.extracted/squashfs-root/
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root]
└──╼ $ls
bin  dev  etc  home  lib  linuxrc  proc  root  sbin  tmp  usr  var
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root]
└──╼ $
```

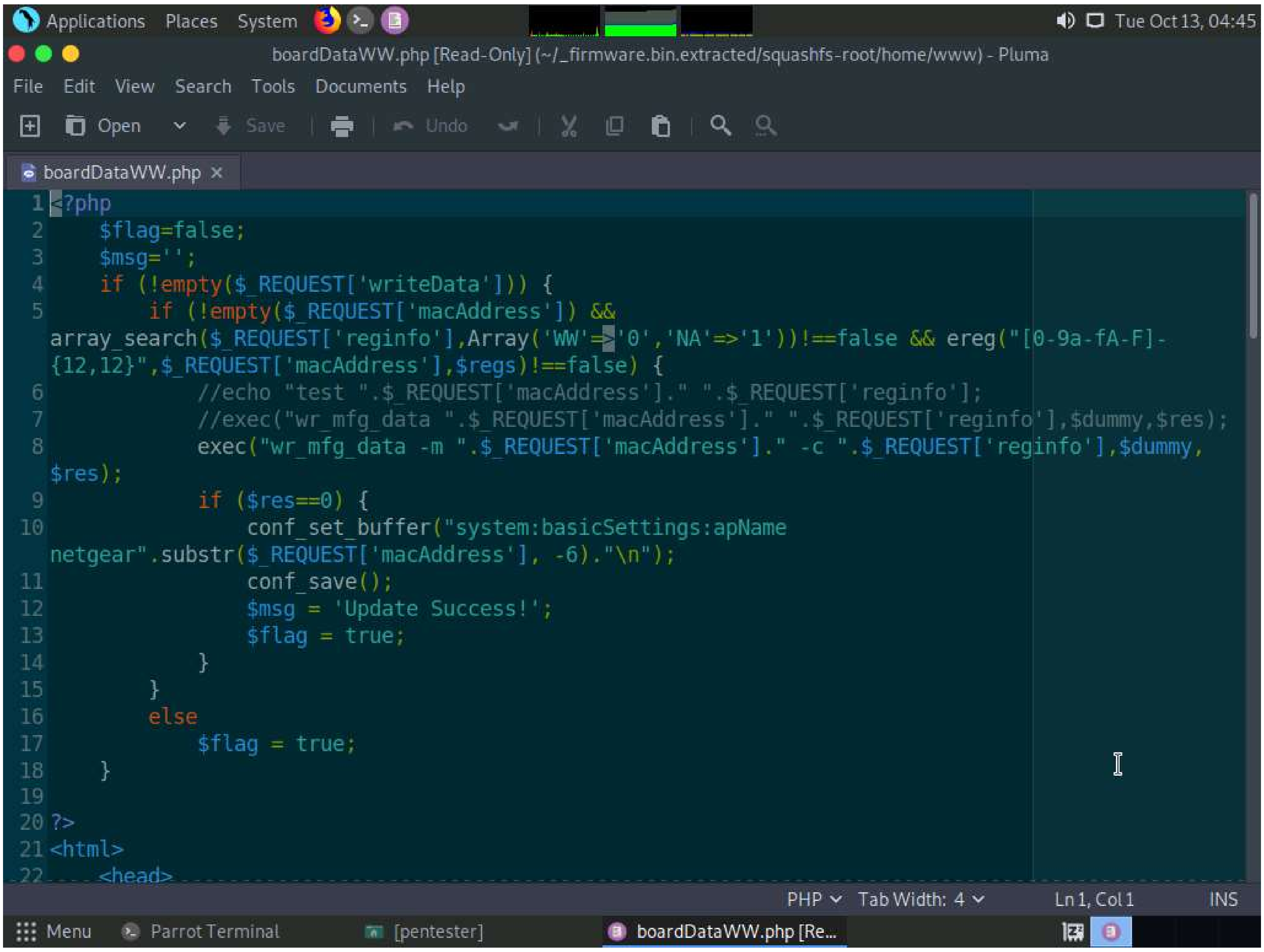
10. From here, you can explore it to see the contents. A good place to start is the home directory since it usually contains information about the installed applications. Type **cd home** and press **Enter**. Type **ls** and press **Enter** to review the results. Since we have a **www** folder, this can be of interest. Type **cd www** and **Enter**, and then type **ls** and press **Enter** to view the file contents.



```
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root]
└──╼ $cd home/
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root/home]
└──╼ $ls
cli  www
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root/home]
└──╼ $cd www/
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root/home/www]
└──╼ $ls
background.html   common.php         include            packetCapture.php  thirdMenu.html
BackupConfig.php  config.php         index.php          recreate.php       thirdMenu.php
boardDataNA.php   data.php           killall.php        redirect.html      titleLogo.php
boardDataWW.php   downloadFile.php   login_button.html  redirect.php       tmpl
body.php          getBoardConfig.php login_header.php   saveTable.php      UserGuide.html
button.html       getJsonData.php    login.php          siteSurvey.php
checkConfig.php   header.php         logout.html        support.link
checkSession.php  help               logout.php         templates
clearLog.php      images             monitorFile.cfg    test.php
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root/home/www]
└──╼ $
```

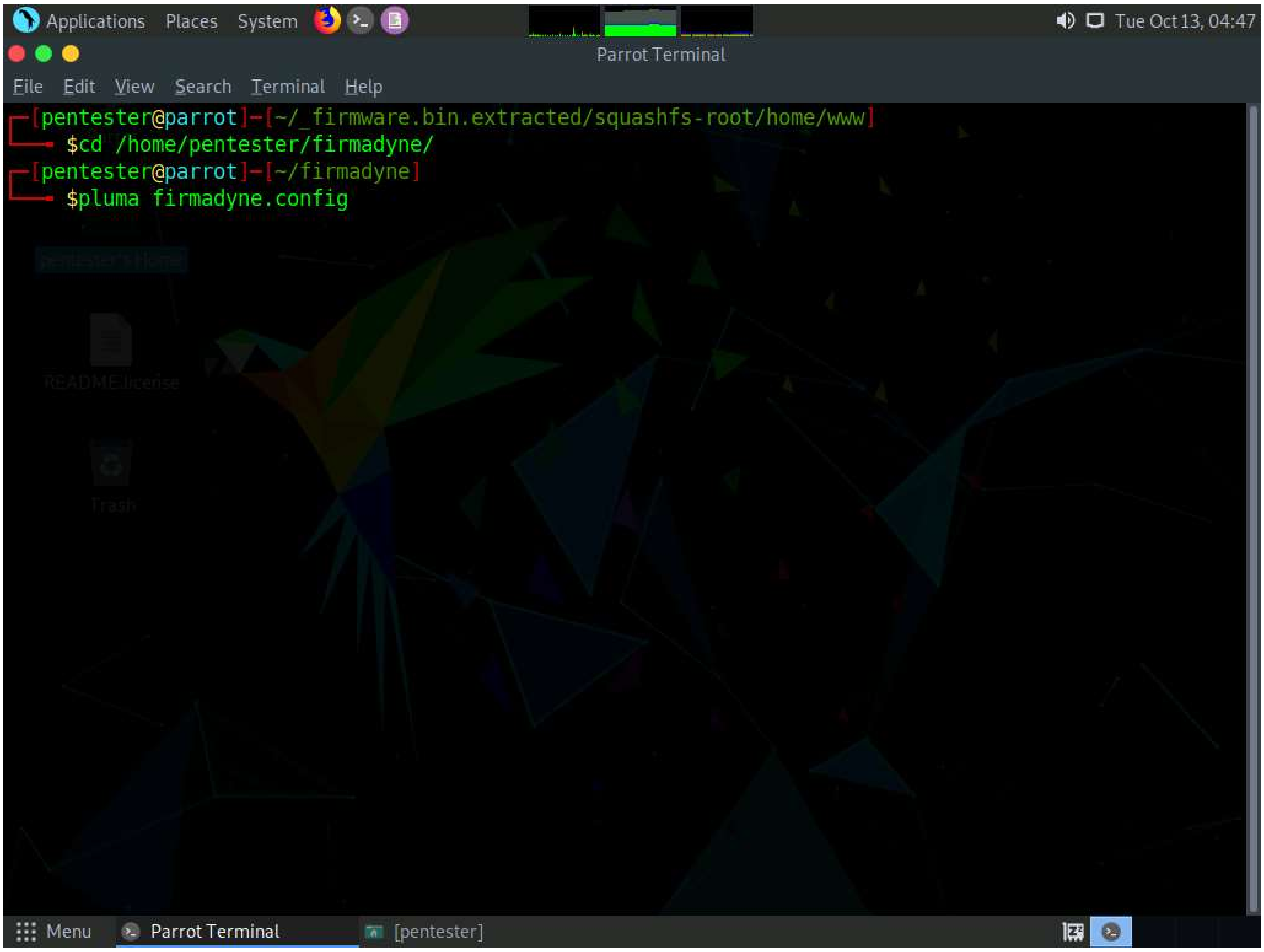
11. We have a lot of code here to examine. In most cases, this code is not written securely. The file we want to look at is the **boardDataWW.php**. Open this in the editor of your choice.

12. As you review the code, it is apparent that there is nothing in the way of input validation and boundary checking. This is a clear indication that this application has injection vulnerabilities. Minimize or close the text editor window. There are many files here for you to explore, but we will leave that to you. Next, we will work on the challenges of mounting and booting the firmware with an emulator.

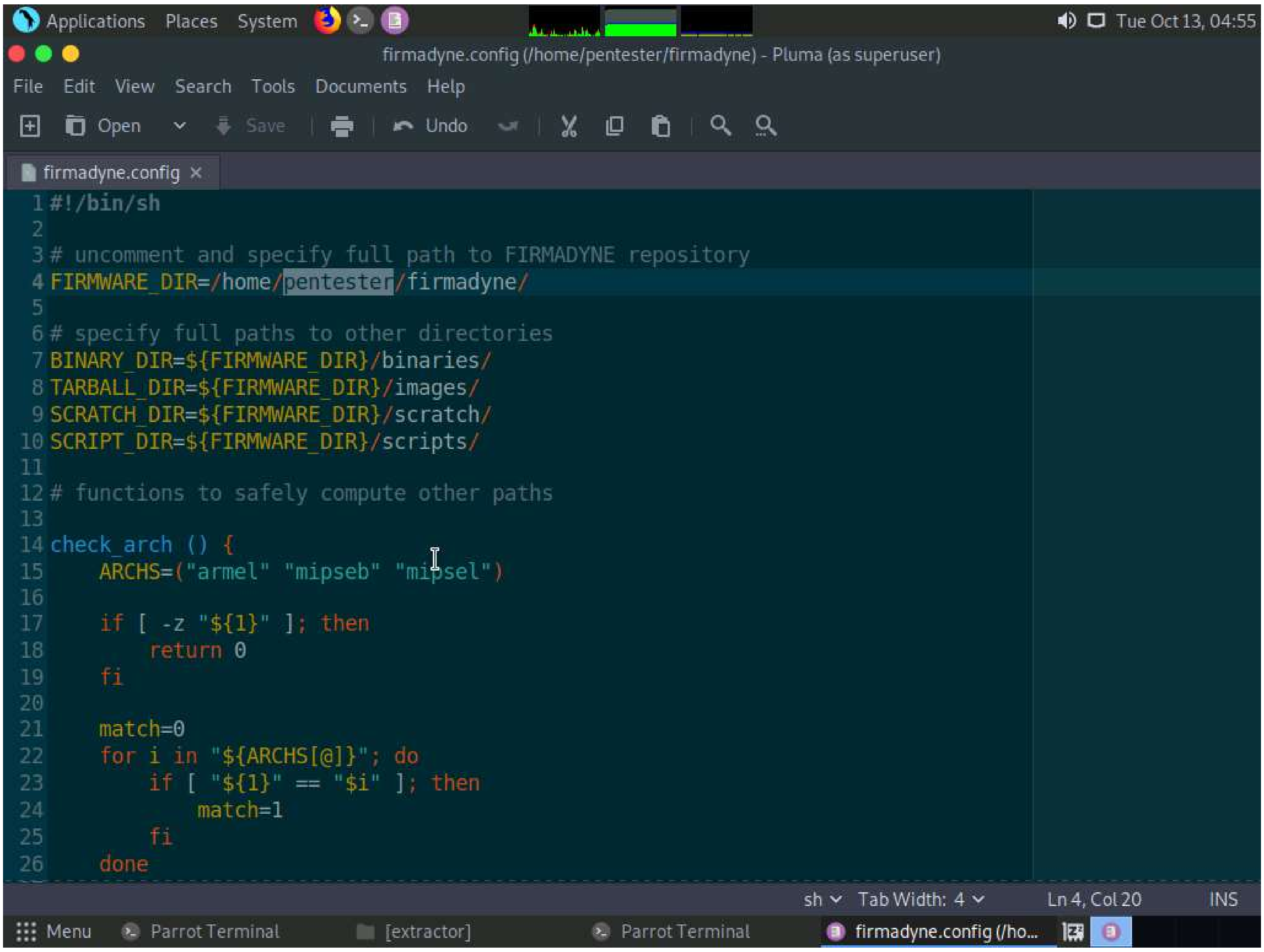

Applications Places System
Tue Oct 13, 04:45
boardDataWW.php [Read-Only] (~/_firmware.bin.extracted/squashfs-root/home/www) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
boardDataWW.php
PHP
Tab Width: 4
Ln 1, Col 1
INS
Menu
Parrot Terminal
[pentester]
boardDataWW.php [Re...

```
<?php
    $flag=false;
    $msg='';
    if (!empty($_REQUEST['writeData'])) {
        if (!empty($_REQUEST['macAddress']) &&
array_search($_REQUEST['reginfo'],Array('WW'=>'0','NA'=>'1'))!==false && ereg("[0-9a-fA-F]-
{12,12}",$_REQUEST['macAddress'],$regs)!==false) {
            //echo "test ".$_REQUEST['macAddress']." ".$_REQUEST['reginfo'];
            //exec("wr_mfg_data ".$_REQUEST['macAddress']." ".$_REQUEST['reginfo'],$dummy,$res);
            exec("wr_mfg_data -m ".$_REQUEST['macAddress']." -c ".$_REQUEST['reginfo'],$dummy,
$res);
            if ($res==0) {
                conf_set_buffer("system:basicSettings:apName
netgear".substr($_REQUEST['macAddress'], -6)."\n");
                conf_save();
                $msg = 'Update Success!';
                $flag = true;
            }
        }
        else
            $flag = true;
    }

?>
<html>
    <head>
```

13. The one configuration we want to make is point the **firmadyne** directory to the correct address. Type **cd /home/pentester/firmadyne** and press **Enter** and open the config file **firmadyne.config** in the text editor of your choice and set **FIRMWARE_DIR** to point to the location of the tool, and **uncomment** the line as shown in the screenshot. Save and close the text editor.



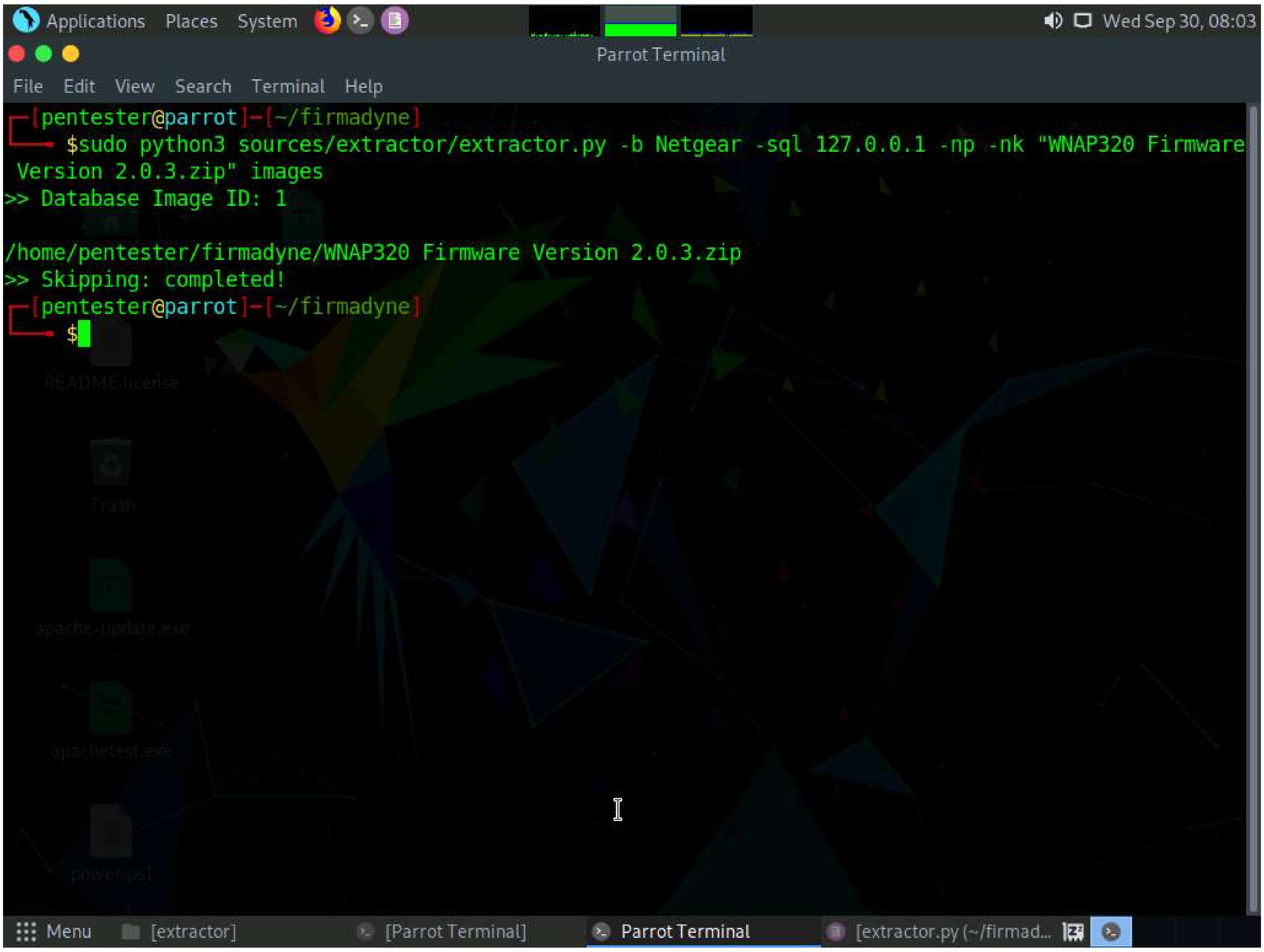
```
┌─[pentester@parrot]─[~/_firmware.bin.extracted/squashfs-root/home/www]
└──╼ $cd /home/pentester/firmadyne/
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $pluma firmadyne.config
```



```
#!/bin/sh

# uncomment and specify full path to FIRMADYNE repository
FIRMWARE_DIR=/home/pentester/firmadyne/

# specify full paths to other directories
BINARY_DIR=${FIRMWARE_DIR}/binaries/
TARBALL_DIR=${FIRMWARE_DIR}/images/
SCRATCH_DIR=${FIRMWARE_DIR}/scratch/
SCRIPT_DIR=${FIRMWARE_DIR}/scripts/

# functions to safely compute other paths

check_arch () {
    ARCHS=("armel" "mipseb" "mipsel")

    if [ -z "${1}" ]; then
        return 0
    fi

    match=0
    for i in "${ARCHS[@]}"; do
        if [ "${1}" == "$i" ]; then
            match=1
        fi
    done
```

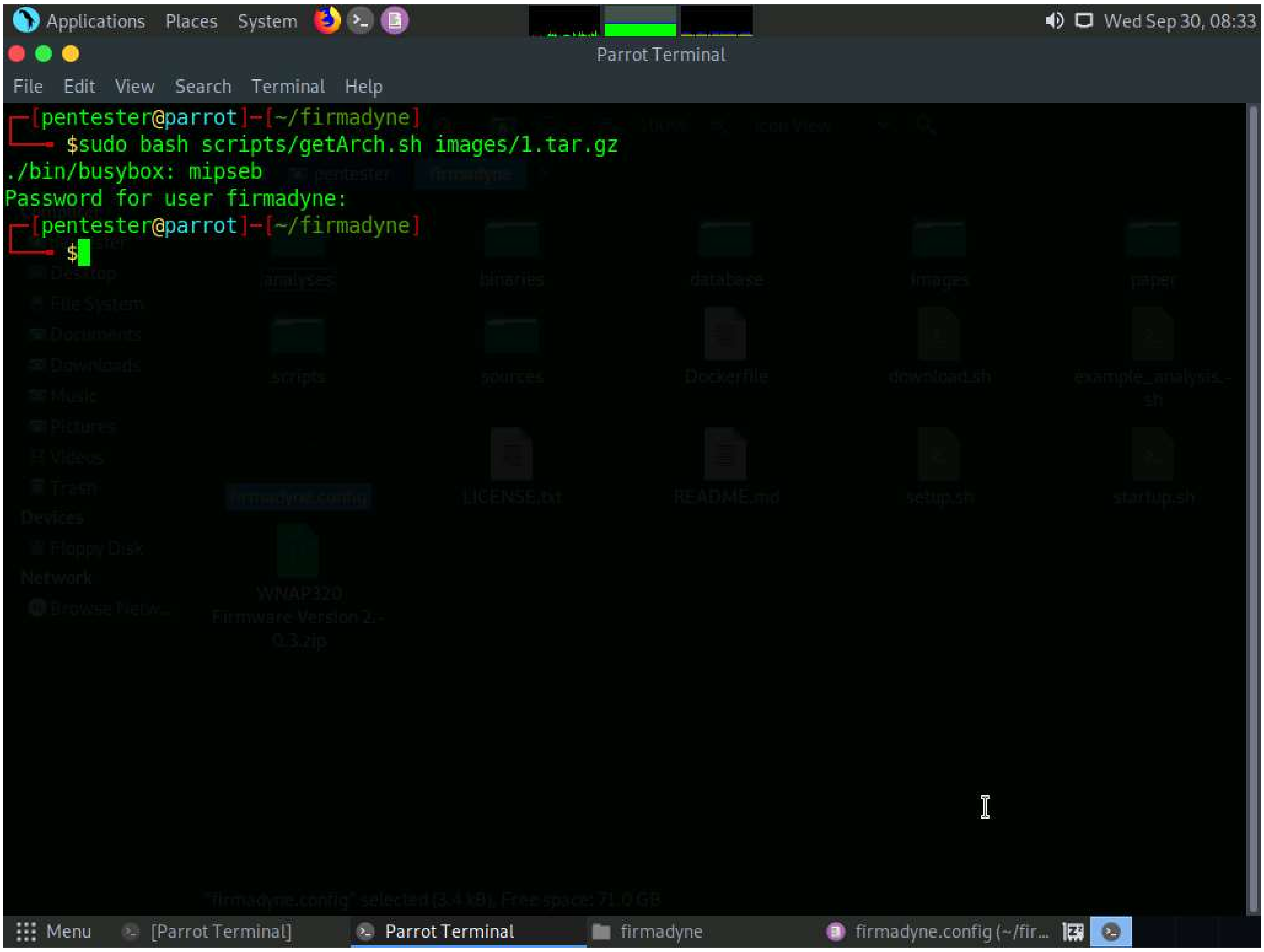
14. We will work with the compressed firmware version straight from the download. This is the same file as **firmware.bin**. Ensure that you are in the firmadyne directory.

15. Use the extractor to recover only the filesystem. Ensure that the following are set: no kernel (-nk), no parallel operation (-np), populating the image table in the SQL server at 127.0.0.1 (-sql) with the Netgear brand (-b), and storing the tarball in images. Type **sudo python3 sources/extractor/extractor.py -b Netgear -sql 127.0.0.1 -np -nk "WNAP320 Firmware Version 2.0.3.zip" images** and press **Enter**.



```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo python3 sources/extractor/extractor.py -b Netgear -sql 127.0.0.1 -np -nk "WNAP320 Firmware
 Version 2.0.3.zip" images
>> Database Image ID: 1

/home/pentester/firmadyne/WNAP320 Firmware Version 2.0.3.zip
>> Skipping: completed!
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

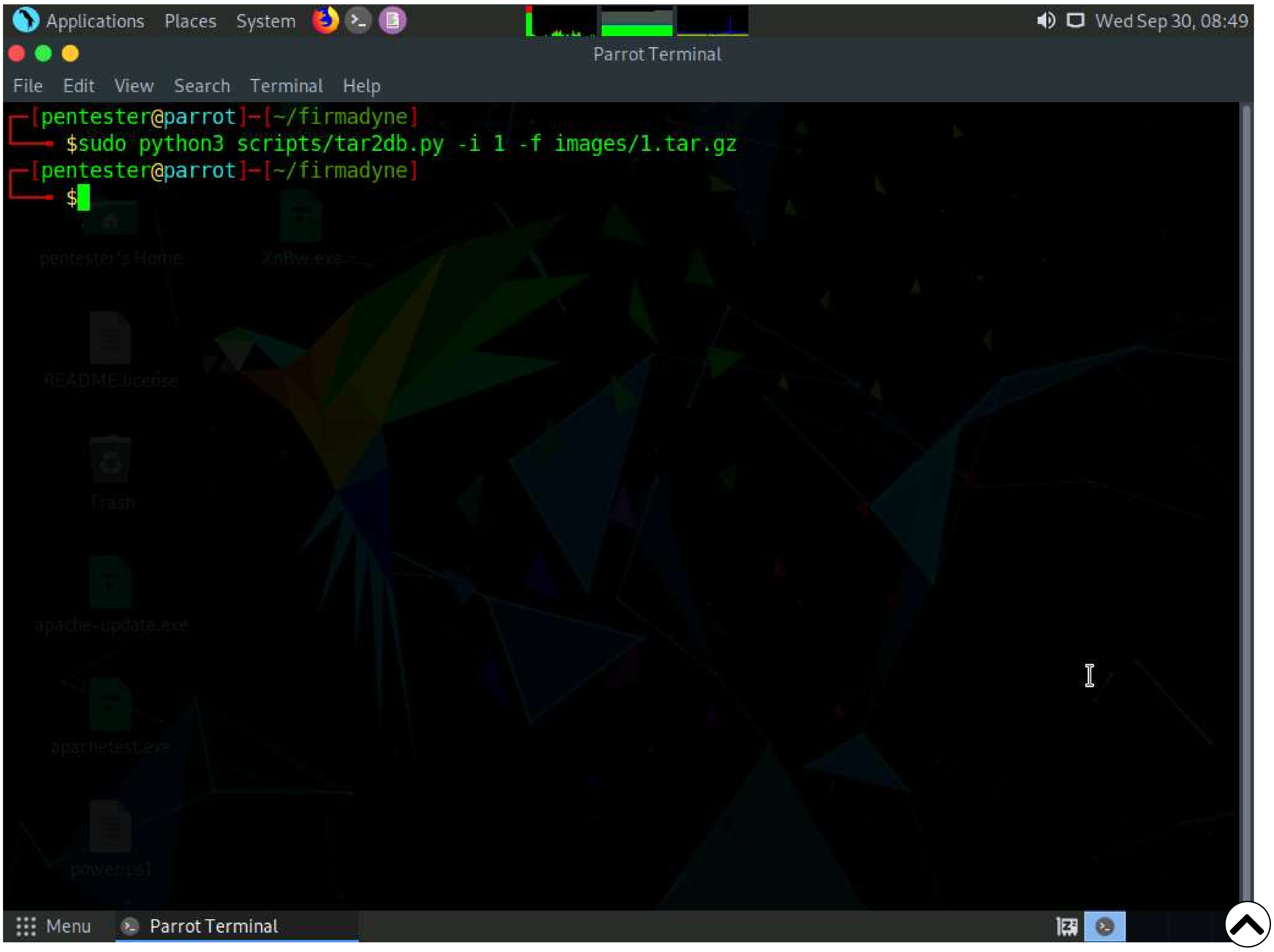
16. If it works correctly, you will have a tarball in the image directory. Next, enter **sudo bash scripts/getArch.sh images/1.tar.gz**. Enter the password of firmadyne which is **firmadyne** when prompted.


Applications Places System
Wed Sep 30, 08:33
Parrot Terminal
File Edit View Search Terminal Help
Menu
[Parrot Terminal]
Parrot Terminal
firmadyne
firmadyne.config (~/fir...

```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scripts/getArch.sh images/1.tar.gz
./bin/busybox: mipseb
Password for user firmadyne:
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

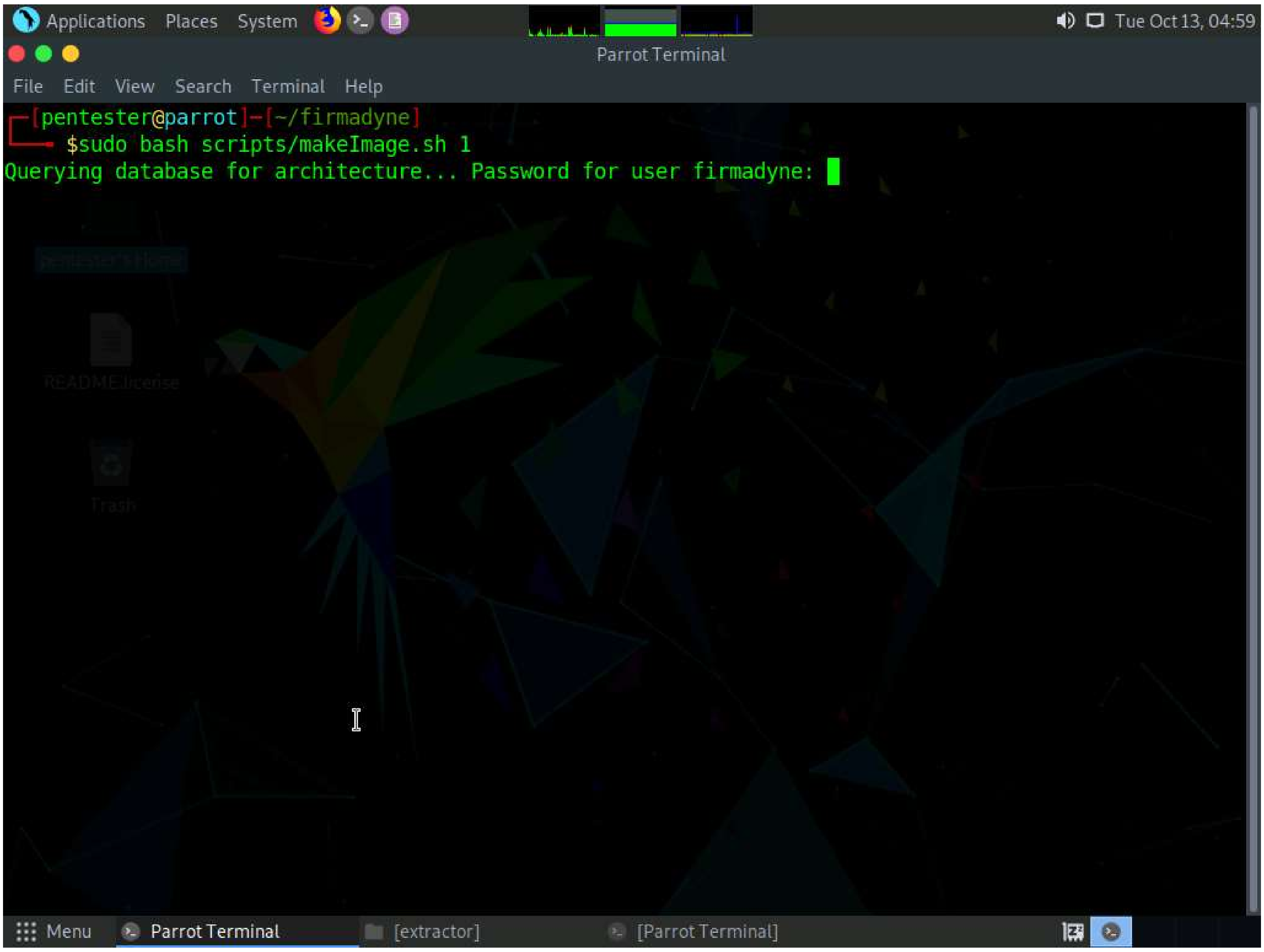
17. This command is used to identify the architecture of firmware **1** and store the result in the image table of the database.

18. Now we want to load the contents of the filesystem for firmware **1** into the database and populate the object and object*to*image tables. Enter **sudo python3 scripts/tar2db.py -i 1 -f images/1.tar.gz**.

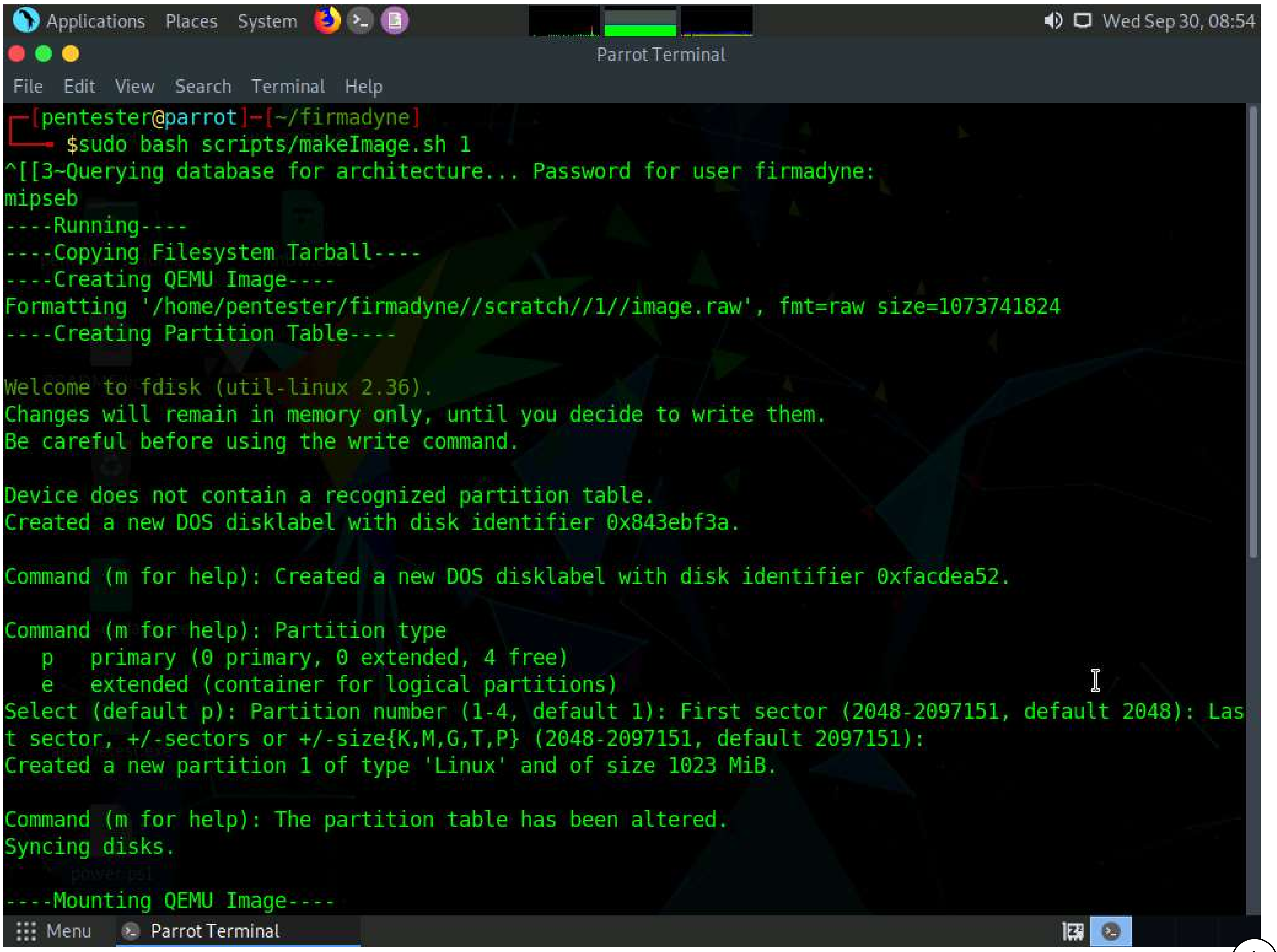

Applications Places System
Wed Sep 30, 08:49
Parrot Terminal
File Edit View Search Terminal Help
Menu
Parrot Terminal

```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo python3 scripts/tar2db.py -i 1 -f images/1.tar.gz
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

19. Next, we need to create the QEMU disk image for firmware **1**. Enter **sudo bash scripts/makeImage.sh 1**. Enter the password of firmadyne which is **firmadyne** when prompted.

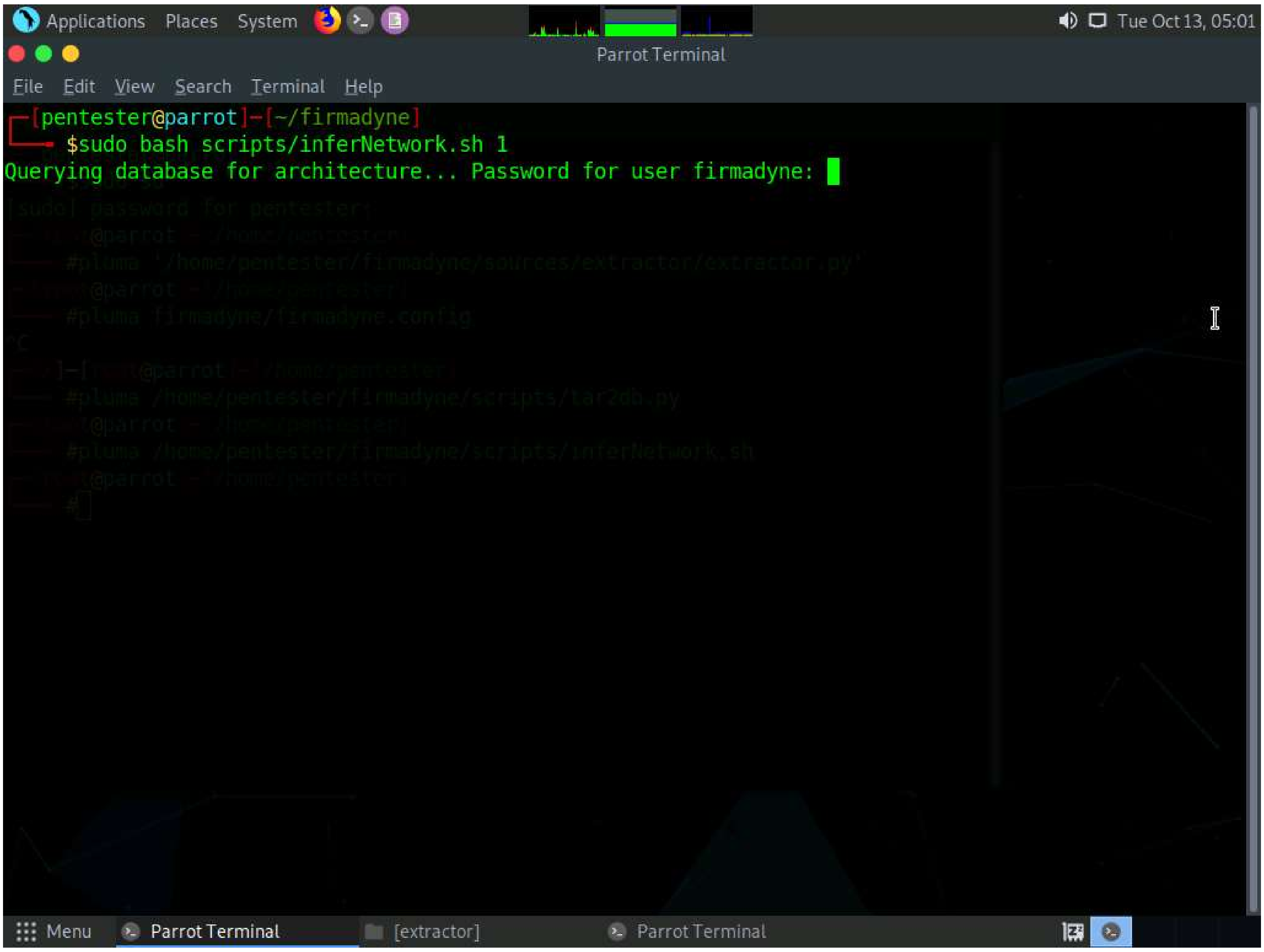


```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scripts/makeImage.sh 1
Querying database for architecture... Password for user firmadyne: 
```

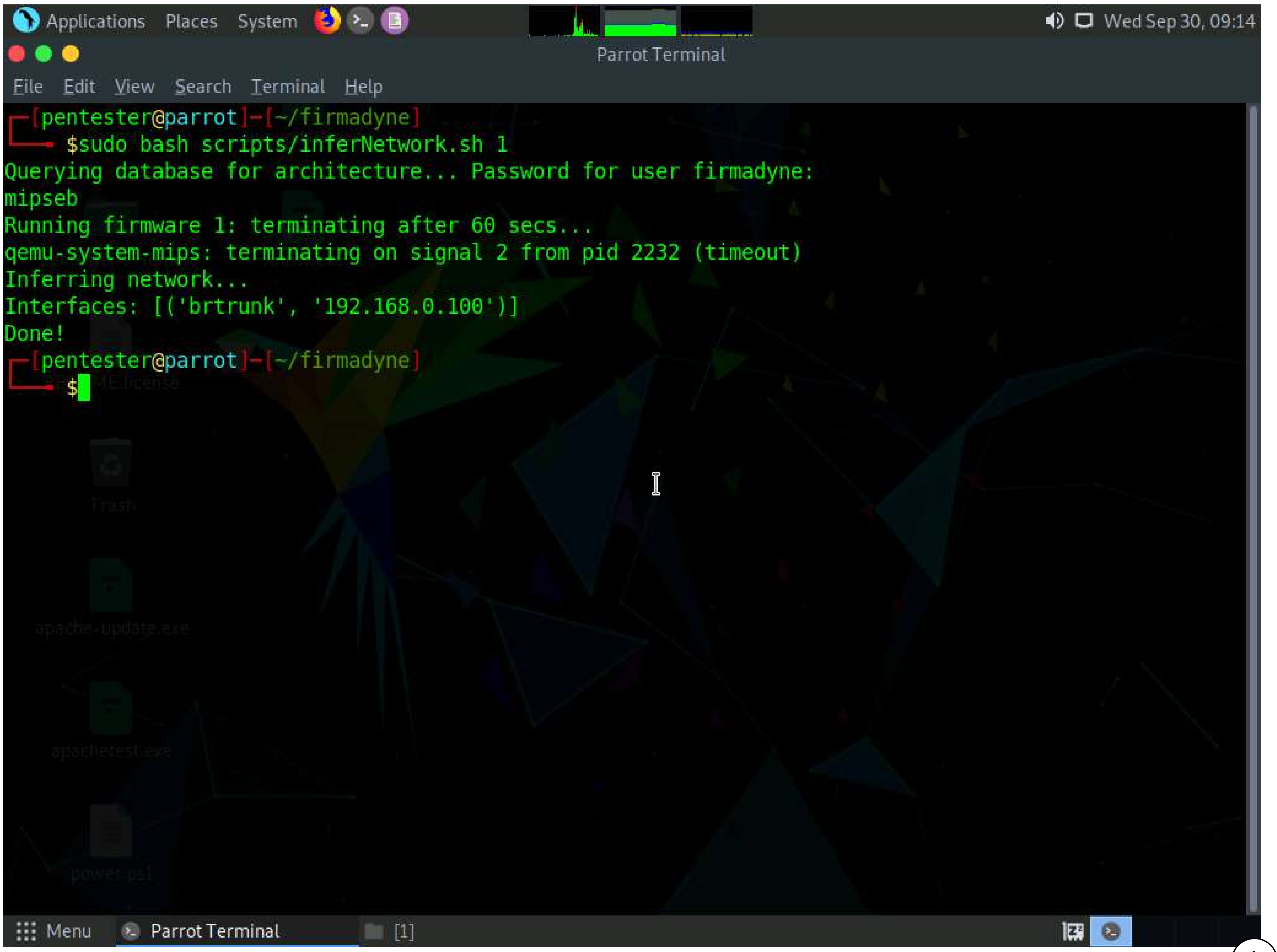


```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scripts/makeImage.sh 1
^[[3~Querying database for architecture... Password for user firmadyne:
mipseb
----Running----
----Copying Filesystem Tarball----
----Creating QEMU Image----
Formatting '/home/pentester/firmadyne//scratch//1//image.raw', fmt=raw size=1073741824
----Creating Partition Table----

Welcome to fdisk (util-linux 2.36).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x843ebf3a.

Command (m for help): Created a new DOS disklabel with disk identifier 0xfacdea52.

Command (m for help): Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): Partition number (1-4, default 1): First sector (2048-2097151, default 2048): Las
t sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151):
Created a new partition 1 of type 'Linux' and of size 1023 MiB.

Command (m for help): The partition table has been altered.
Syncing disks.

----Mounting QEMU Image----
```

20. Now you are ready to setup networking. It should then be available on the network like any other node. Enter **sudo bash scripts/inferNetwork.sh 1**. Enter the password of firmadyne i.e., **firmadyne** when prompted.

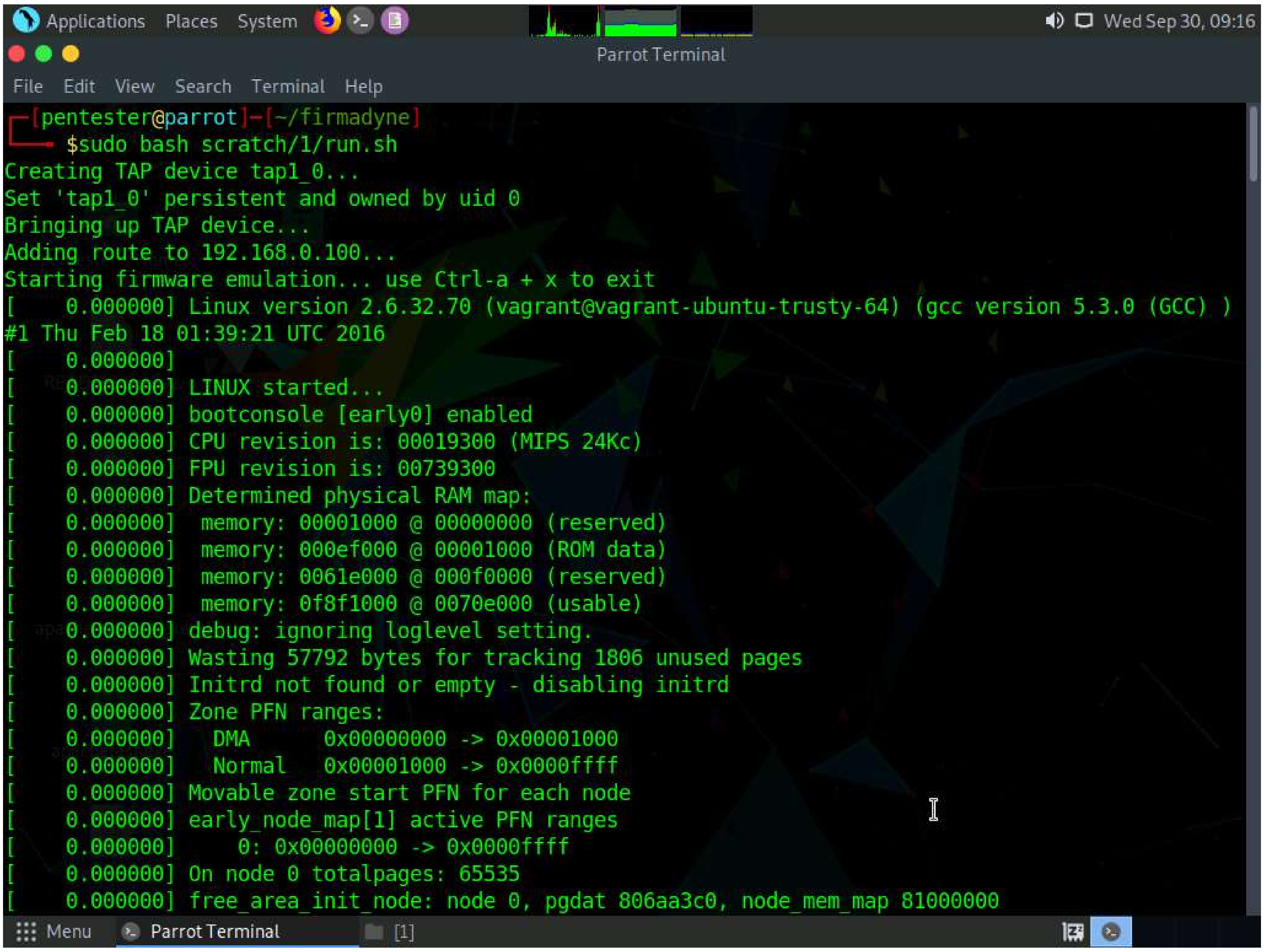


```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scripts/inferNetwork.sh 1
Querying database for architecture... Password for user firmadyne:
```



```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scripts/inferNetwork.sh 1
Querying database for architecture... Password for user firmadyne:
mipseb
Running firmware 1: terminating after 60 secs...
qemu-system-mips: terminating on signal 2 from pid 2232 (timeout)
Inferring network...
Interfaces: [('brtrunk', '192.168.0.100')]
Done!
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

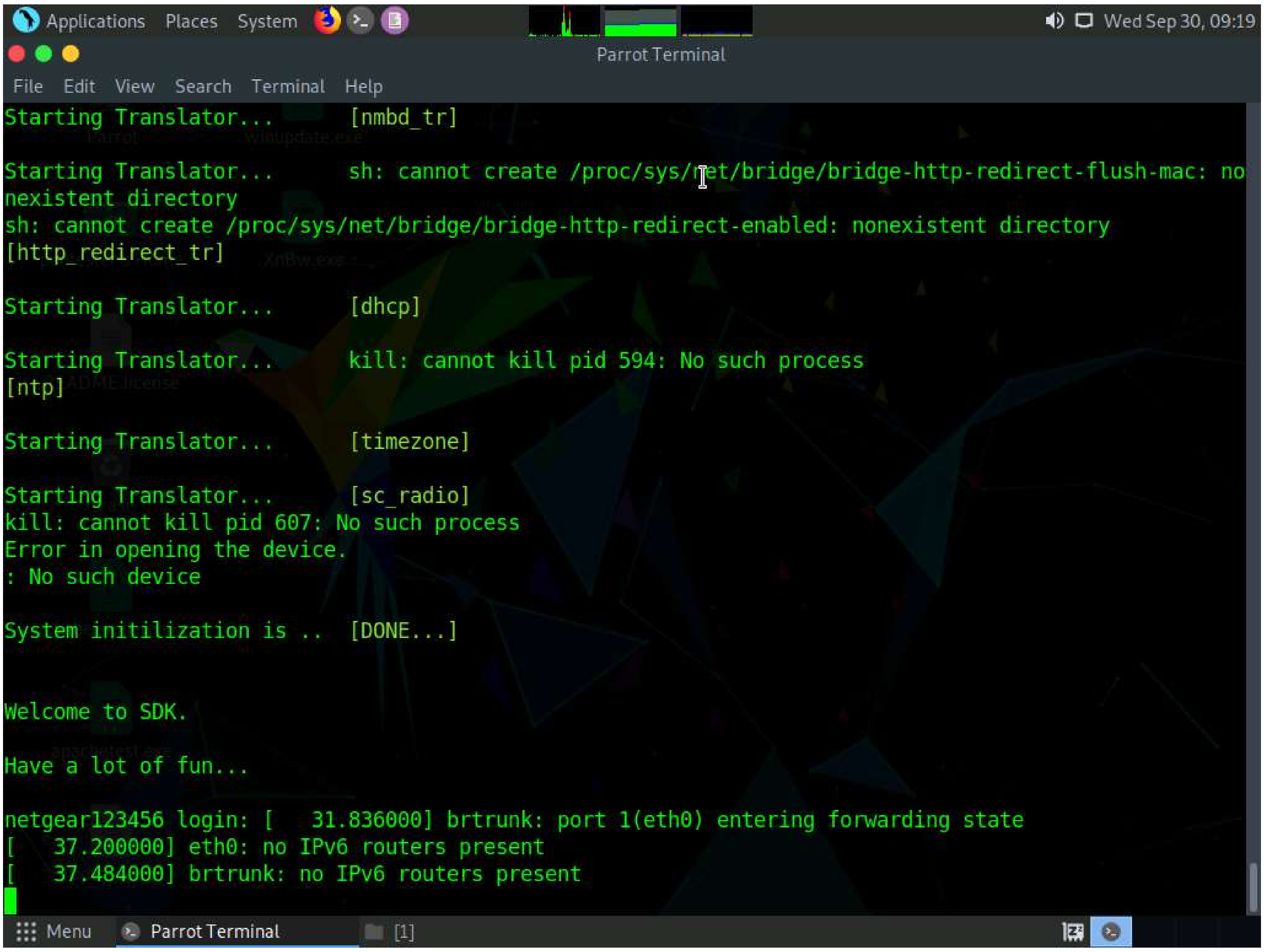
21. Now we are ready to emulate firmware **1** with the inferred network configuration. This will modify the configuration of the host system by creating a Test Access Point (TAP) device and adding a route. Enter **sudo bash scratch/1/run.sh**.



Applications Places System
Wed Sep 30, 09:16
Parrot Terminal
File Edit View Search Terminal Help
Menu
Parrot Terminal
[1]

```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash scratch/1/run.sh
Creating TAP device tap1_0...
Set 'tap1_0' persistent and owned by uid 0
Bringing up TAP device...
Adding route to 192.168.0.100...
Starting firmware emulation... use Ctrl-a + x to exit
[    0.000000] Linux version 2.6.32.70 (vagrant@vagrant-ubuntu-trusty-64) (gcc version 5.3.0 (GCC) )
#1 Thu Feb 18 01:39:21 UTC 2016
[    0.000000]
[    0.000000] LINUX started...
[    0.000000] bootconsole [early0] enabled
[    0.000000] CPU revision is: 00019300 (MIPS 24Kc)
[    0.000000] FPU revision is: 00739300
[    0.000000] Determined physical RAM map:
[    0.000000]  memory: 00001000 @ 00000000 (reserved)
[    0.000000]  memory: 000ef000 @ 00001000 (ROM data)
[    0.000000]  memory: 0061e000 @ 000f0000 (reserved)
[    0.000000]  memory: 0f8f1000 @ 0070e000 (usable)
[    0.000000] debug: ignoring loglevel setting.
[    0.000000] Wasting 57792 bytes for tracking 1806 unused pages
[    0.000000] Initrd not found or empty - disabling initrd
[    0.000000] Zone PFN ranges:
[    0.000000]   DMA      0x00000000 -> 0x00001000
[    0.000000]   Normal   0x00001000 -> 0x0000ffff
[    0.000000] Movable zone start PFN for each node
[    0.000000] early_node_map[1] active PFN ranges
[    0.000000]     0: 0x00000000 -> 0x0000ffff
[    0.000000] On node 0 totalpages: 65535
[    0.000000] free_area_init_node: node 0, pgdat 806aa3c0, node_mem_map 81000000
```

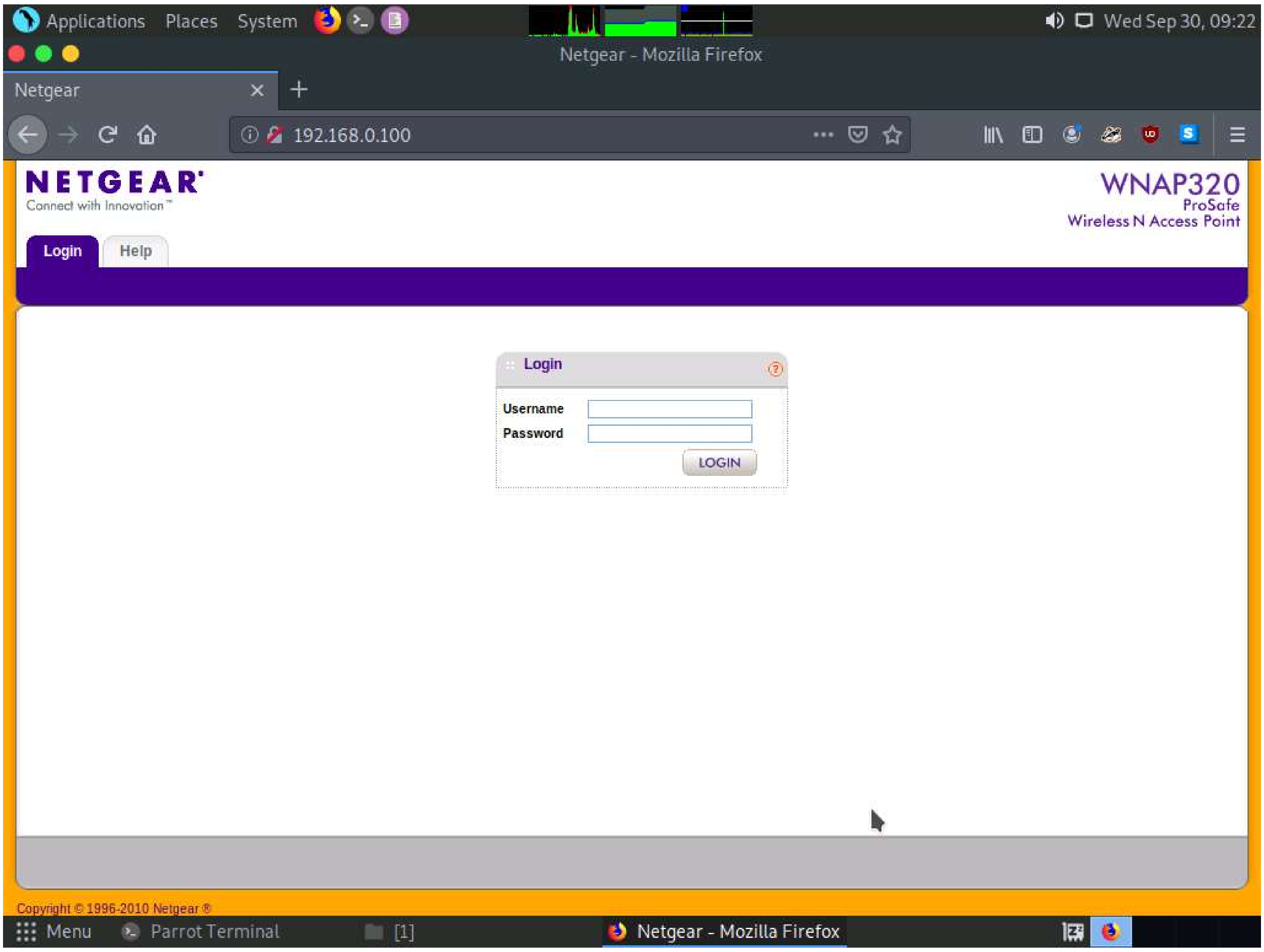
22. This will take some time and execute a chain of instructions as shown in the screenshot. Minimize the terminal window.



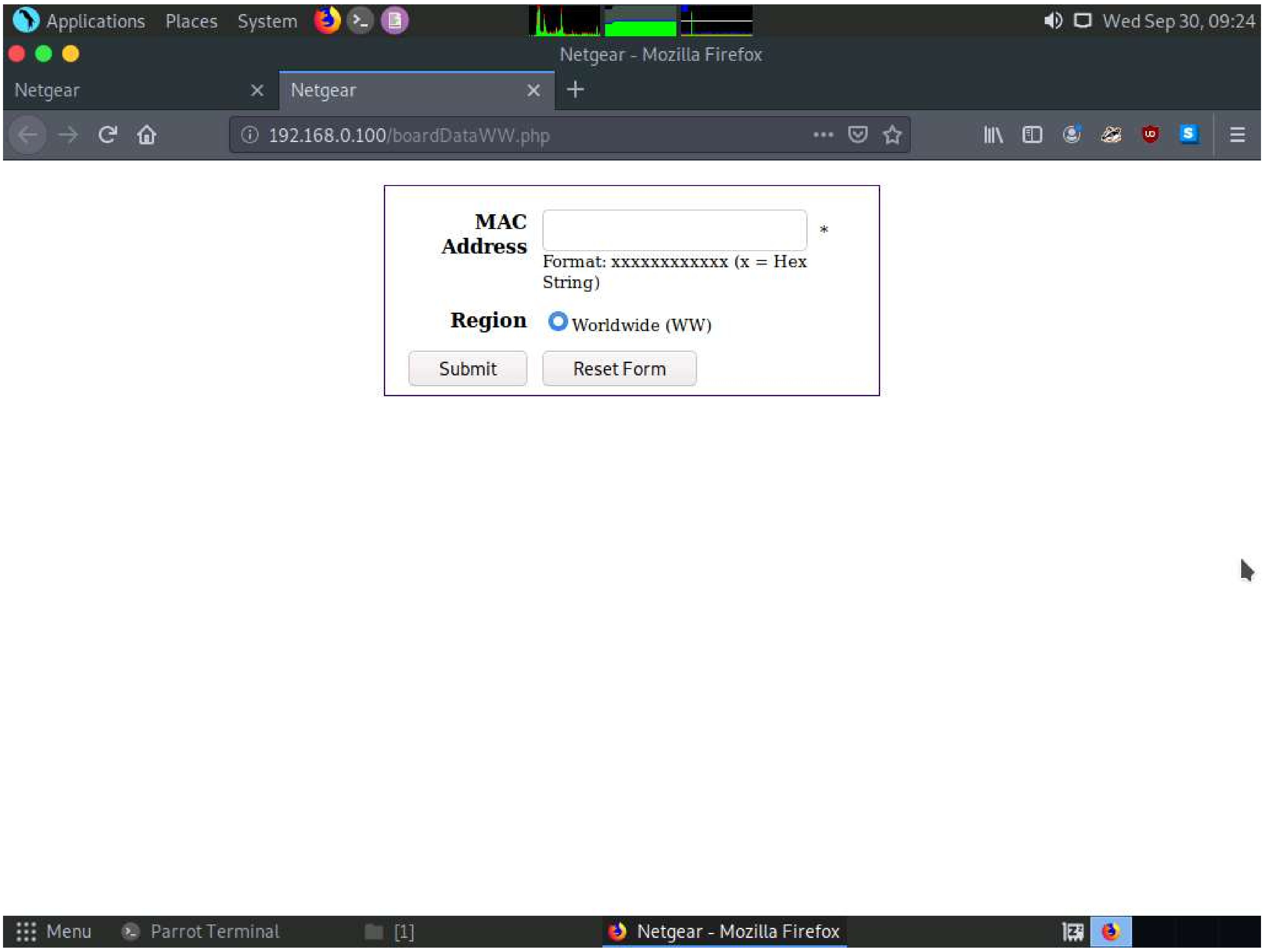
Applications Places System
Wed Sep 30, 09:19
Parrot Terminal
File Edit View Search Terminal Help
Menu Parrot Terminal [1]

```
Starting Translator...       [nmbd_tr]

Starting Translator...       sh: cannot create /proc/sys/net/bridge/bridge-http-redirect-flush-mac: no
nexistent directory
sh: cannot create /proc/sys/net/bridge/bridge-http-redirect-enabled: nonexistent directory
[http_redirect_tr]

Starting Translator...       [dhcp]

Starting Translator...       kill: cannot kill pid 594: No such process
[ntp]

Starting Translator...       [timezone]

Starting Translator...       [sc_radio]
kill: cannot kill pid 607: No such process
Error in opening the device.
: No such device

System initilization is ..   [DONE...]


Welcome to SDK.

Have a lot of fun...

netgear123456 login: [   31.836000] brtrunk: port 1(eth0) entering forwarding state
[   37.200000] eth0: no IPv6 routers present
[   37.484000] brtrunk: no IPv6 routers present
```

23. The system should be available over the network. Open a browser, and enter **http://192.168.0.100**. Netgear WNAP320 access point login page should appear as shown in the screenshot. This is the web interface from the running firmware that is being emulated.



24. We can next look at any of the PHP code that we saw when we explored the file system. In the browser, enter http://192.168.0.100/boardDataWW.php. This will bring up another form as shown in the following screenshot.


Applications Places System
Wed Sep 30, 09:24
Netgear - Mozilla Firefox
Netgear
Netgear
192.168.0.100/boardDataWW.php
Menu
Parrot Terminal
[1]
Netgear - Mozilla Firefox

MAC Address *

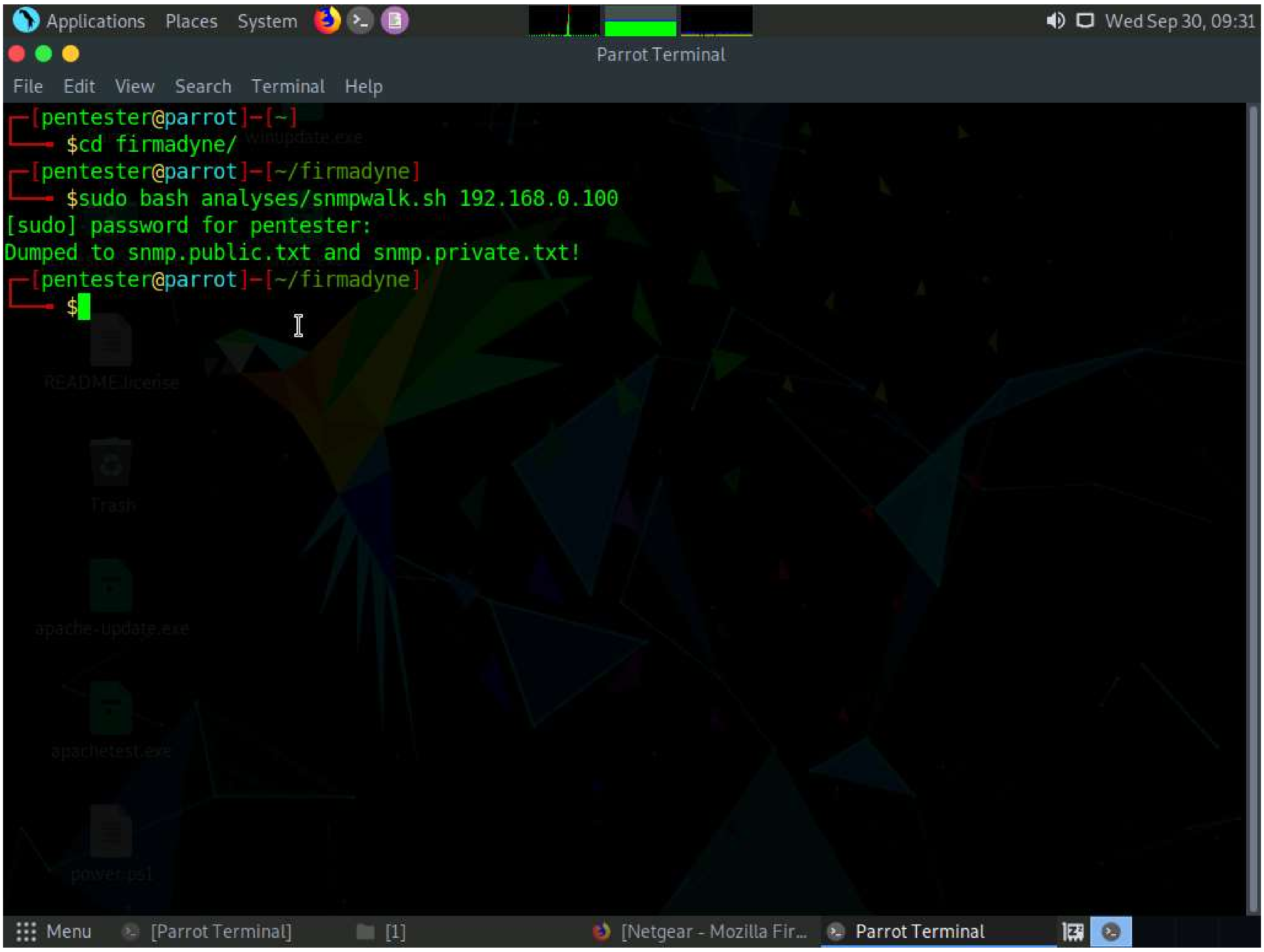
Format: xxxxxxxxxxxx (x = Hex String)

Region ● Worldwide (WW)

Submit | Reset Form

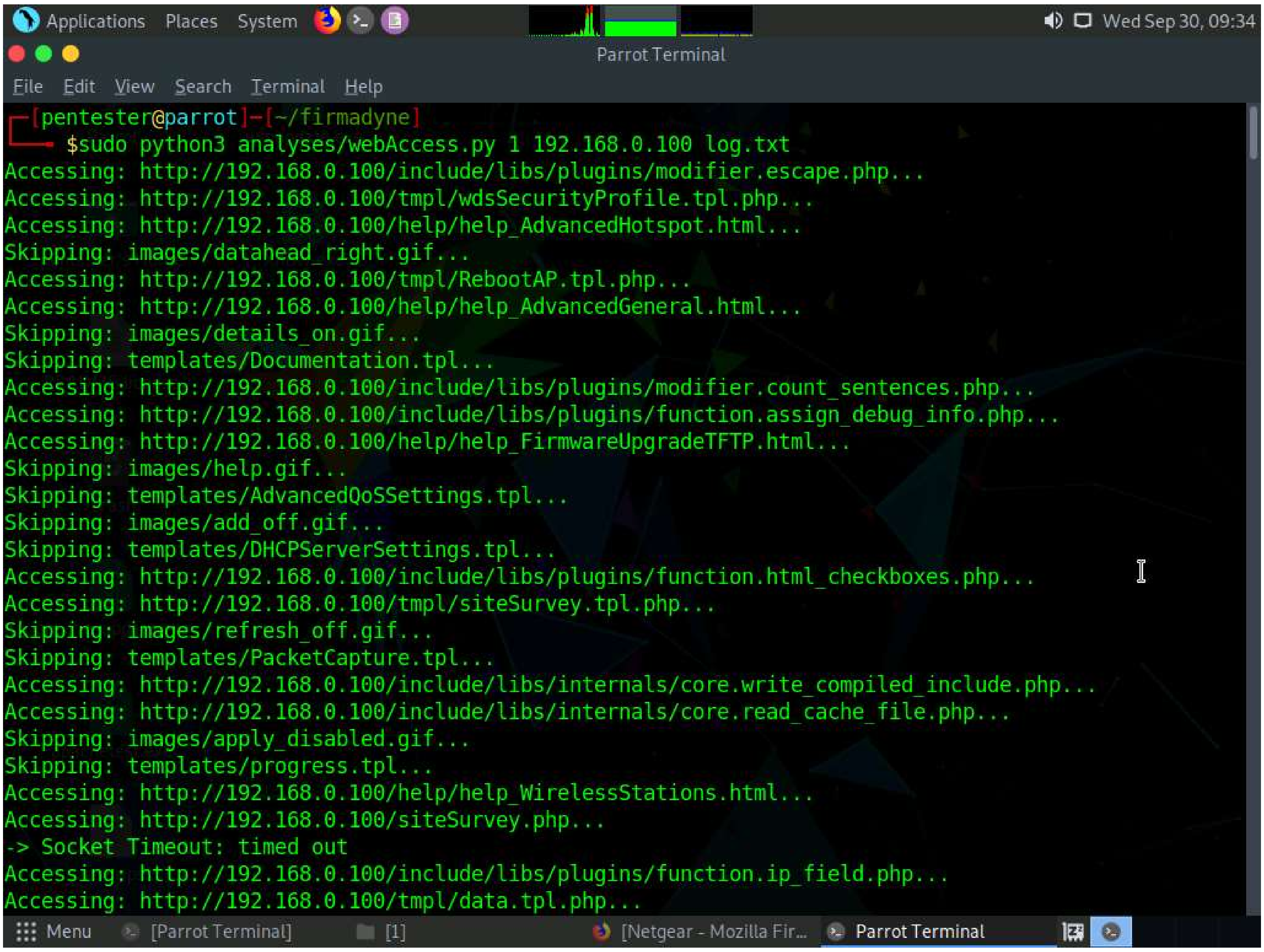
25. Since we had the advantage of reviewing the code, we know that we do not have any input validation, which means that there is a possibility of command injection. We will leave that for you to explore. Hint: You can intercept the query using Burpsuite or any other proxy, and then attempt to enter commands such as cat /etc/passwd.

26. Within the Firmadyne tool, we can leverage additional scripts. Open a new terminal and change the directory to **firmadyne**. Now, enter **sudo bash analyses/snmpwalk.sh 192.168.0.100**. As the command states, it will enumerate data using SNMP. If you are asked to enter the password, type **toor** and press **Enter**.

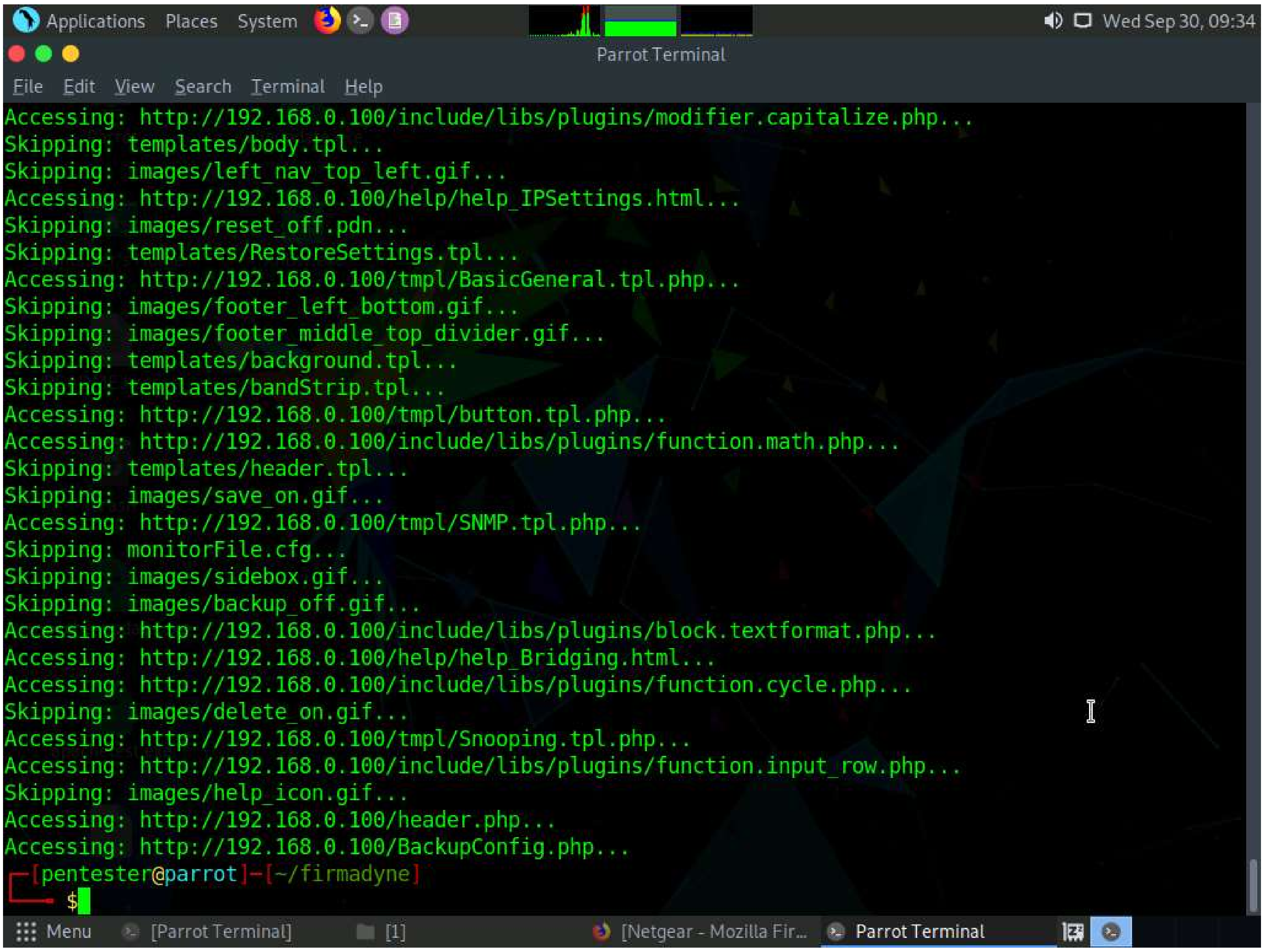


```
┌─[pentester@parrot]─[~]
└──╼ $cd firmadyne/
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo bash analyses/snmpwalk.sh 192.168.0.100
[sudo] password for pentester:
Dumped to snmp.public.txt and snmp.private.txt!
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

27. We can next enter **sudo python3 analyses/webAccess.py 1 192.168.0.100 log.txt**.

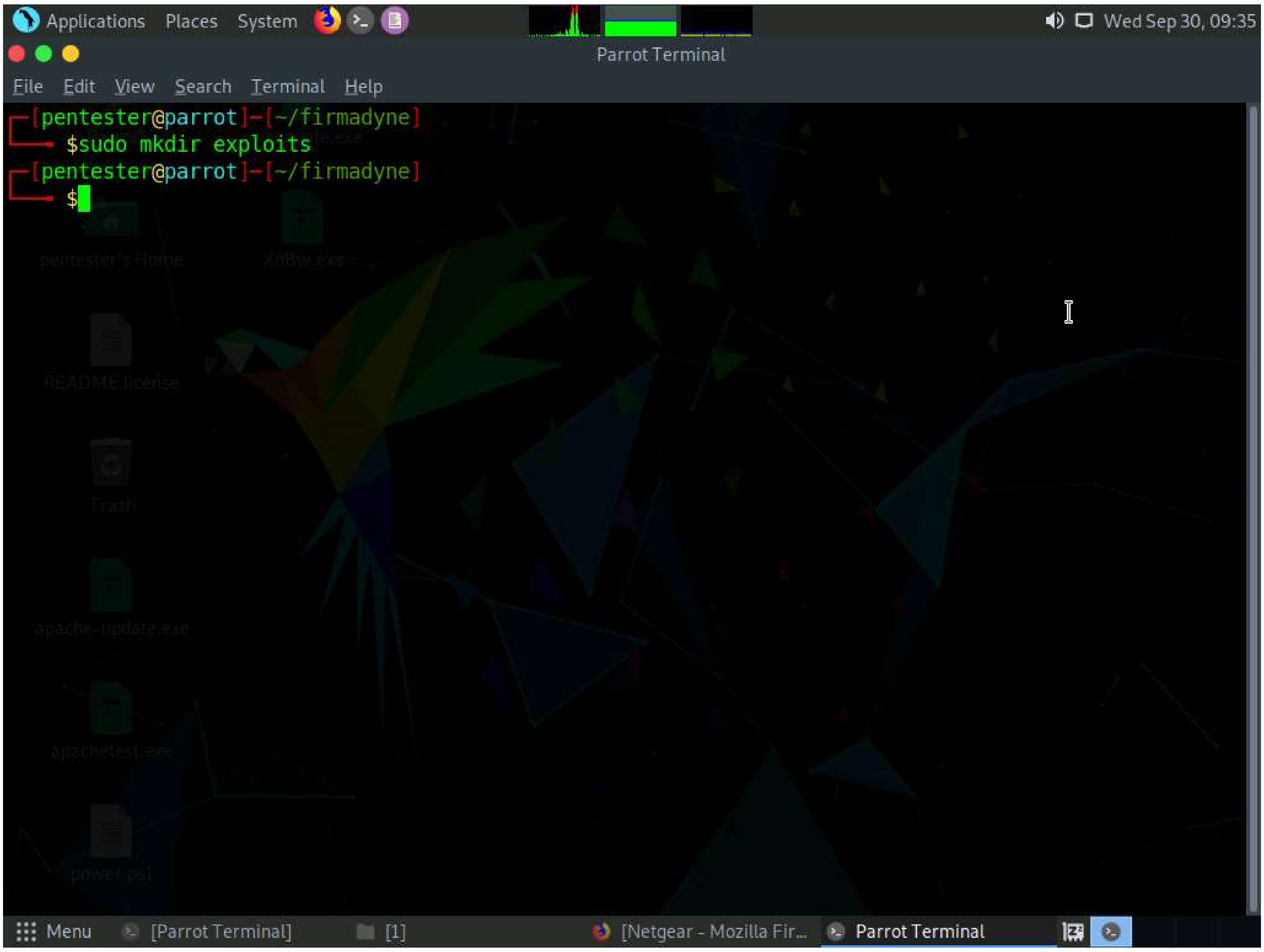


```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo python3 analyses/webAccess.py 1 192.168.0.100 log.txt
Accessing: http://192.168.0.100/include/libs/plugins/modifier.escape.php...
Accessing: http://192.168.0.100/tmpl/wdsSecurityProfile.tpl.php...
Accessing: http://192.168.0.100/help/help_AdvancedHotspot.html...
Skipping: images/datahead_right.gif...
Accessing: http://192.168.0.100/tmpl/RebootAP.tpl.php...
Accessing: http://192.168.0.100/help/help_AdvancedGeneral.html...
Skipping: images/details_on.gif...
Skipping: templates/Documentation.tpl...
Accessing: http://192.168.0.100/include/libs/plugins/modifier.count_sentences.php...
Accessing: http://192.168.0.100/include/libs/plugins/function.assign_debug_info.php...
Accessing: http://192.168.0.100/help/help_FirmwareUpgradeTFTP.html...
Skipping: images/help.gif...
Skipping: templates/AdvancedQoSSettings.tpl...
Skipping: images/add_off.gif...
Skipping: templates/DHCPServerSettings.tpl...
Accessing: http://192.168.0.100/include/libs/plugins/function.html_checkboxes.php...
Accessing: http://192.168.0.100/tmpl/siteSurvey.tpl.php...
Skipping: images/refresh_off.gif...
Skipping: templates/PacketCapture.tpl...
Accessing: http://192.168.0.100/include/libs/internals/core.write_compiled_include.php...
Accessing: http://192.168.0.100/include/libs/internals/core.read_cache_file.php...
Skipping: images/apply_disabled.gif...
Skipping: templates/progress.tpl...
Accessing: http://192.168.0.100/help/help_WirelessStations.html...
Accessing: http://192.168.0.100/siteSurvey.php...
-> Socket Timeout: timed out
Accessing: http://192.168.0.100/include/libs/plugins/function.ip_field.php...
Accessing: http://192.168.0.100/tmpl/data.tpl.php...
```



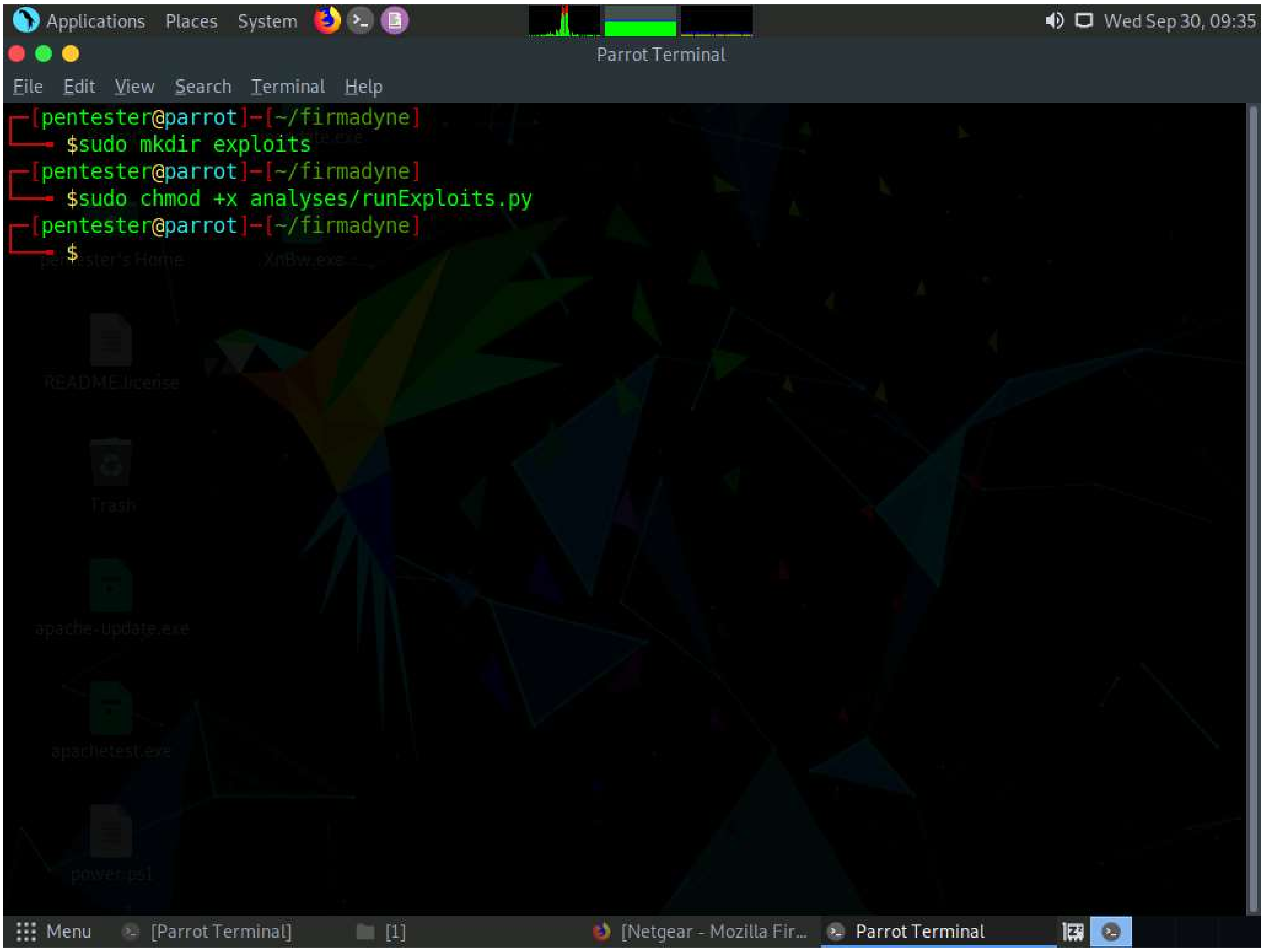
```
Accessing: http://192.168.0.100/include/libs/plugins/modifier.capitalize.php...
Skipping: templates/body.tpl...
Skipping: images/left_nav_top_left.gif...
Accessing: http://192.168.0.100/help/help_IPSettings.html...
Skipping: images/reset_off.pdn...
Skipping: templates/RestoreSettings.tpl...
Accessing: http://192.168.0.100/tmpl/BasicGeneral.tpl.php...
Skipping: images/footer_left_bottom.gif...
Skipping: images/footer_middle_top_divider.gif...
Skipping: templates/background.tpl...
Skipping: templates/bandStrip.tpl...
Accessing: http://192.168.0.100/tmpl/button.tpl.php...
Accessing: http://192.168.0.100/include/libs/plugins/function.math.php...
Skipping: templates/header.tpl...
Skipping: images/save_on.gif...
Accessing: http://192.168.0.100/tmpl/SNMP.tpl.php...
Skipping: monitorFile.cfg...
Skipping: images/sidebox.gif...
Skipping: images/backup_off.gif...
Accessing: http://192.168.0.100/include/libs/plugins/block.textformat.php...
Accessing: http://192.168.0.100/help/help_Bridging.html...
Accessing: http://192.168.0.100/include/libs/plugins/function.cycle.php...
Skipping: images/delete_on.gif...
Accessing: http://192.168.0.100/tmpl/Snooping.tpl.php...
Accessing: http://192.168.0.100/include/libs/plugins/function.input_row.php...
Skipping: images/help_icon.gif...
Accessing: http://192.168.0.100/header.php...
Accessing: http://192.168.0.100/BackupConfig.php...
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

28. There is also a Metasploit interface that will look for vulnerabilities and corresponding exploits. Enter **sudo mkdir exploits**.



```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo mkdir exploits
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

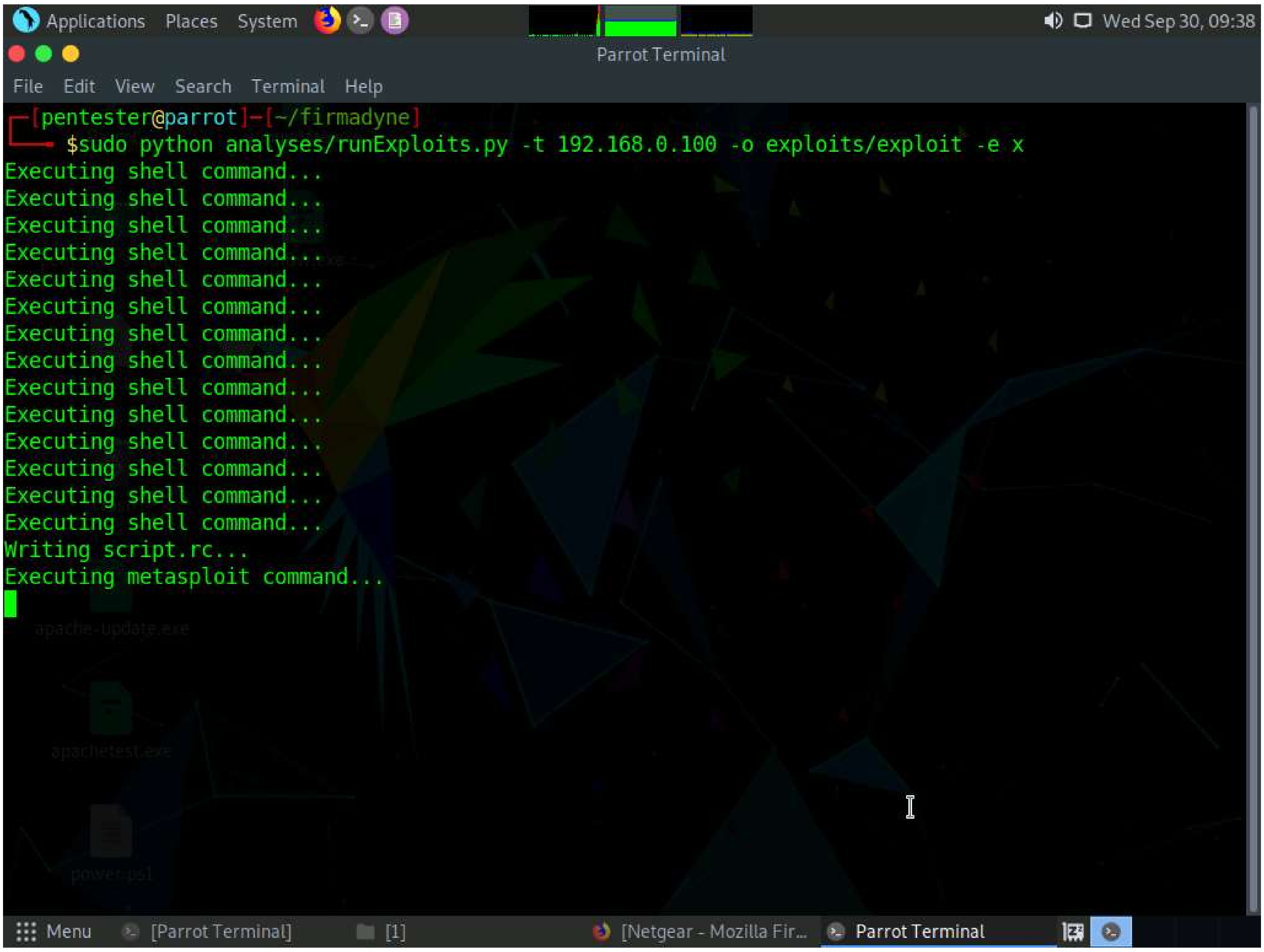
29. Change the permissions on the file with **sudo chmod +x analyses/runExploits.py**.



Applications Places System
Wed Sep 30, 09:35
Parrot Terminal
File Edit View Search Terminal Help
Menu [Parrot Terminal] [1] [Netgear - Mozilla Fir... Parrot Terminal

```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo mkdir exploits
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo chmod +x analyses/runExploits.py
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $
```

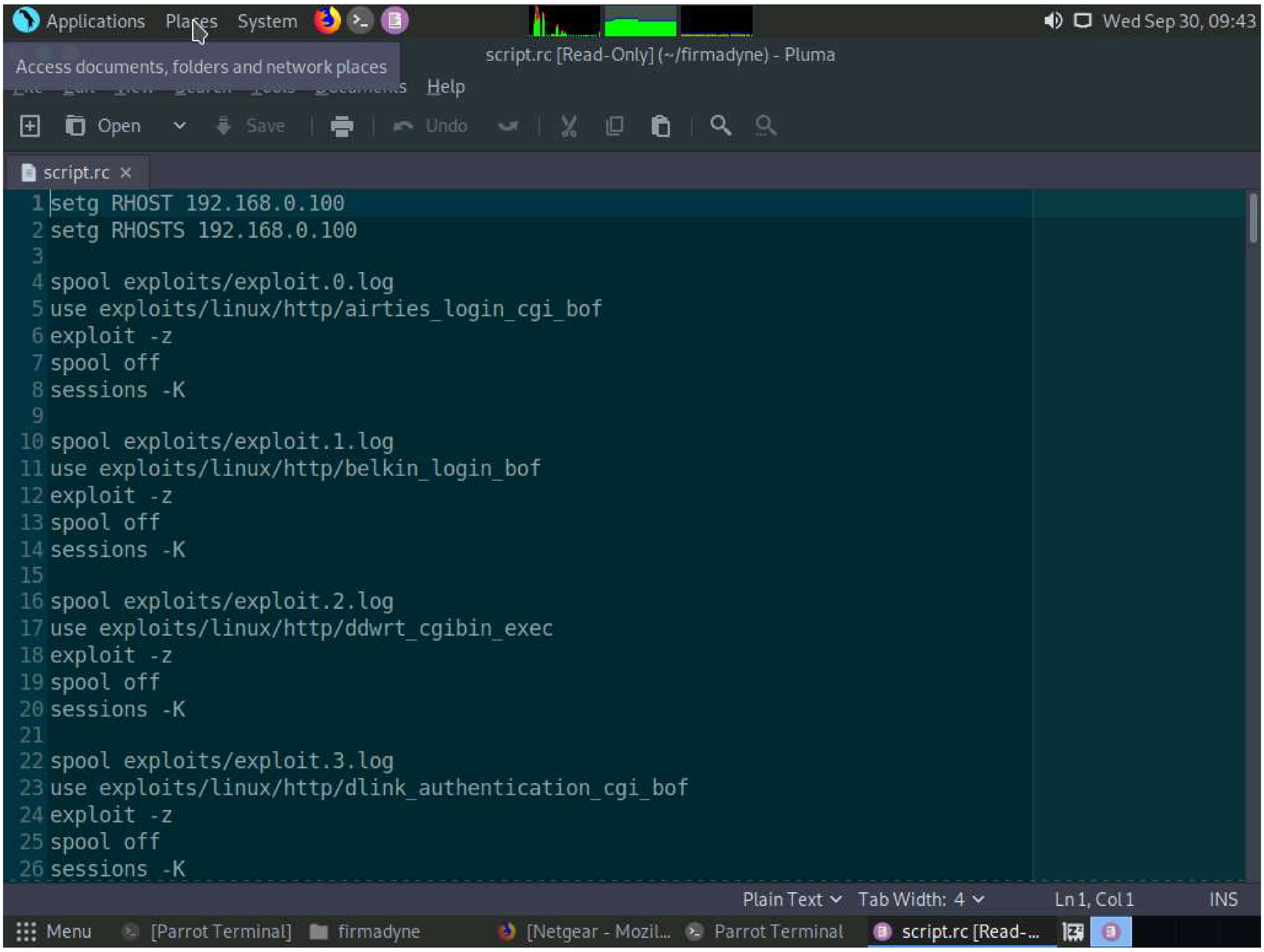
30. Enter **sudo python analyses/runExploits.py -t 192.168.0.100 -o exploits/exploit -e x**. An example of the output of the command is shown in the following screenshot:



Applications Places System
Wed Sep 30, 09:38
Parrot Terminal
File Edit View Search Terminal Help
Menu [Parrot Terminal] [1] [Netgear - Mozilla Fir... Parrot Terminal

```
┌─[pentester@parrot]─[~/firmadyne]
└──╼ $sudo python analyses/runExploits.py -t 192.168.0.100 -o exploits/exploit -e x
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Executing shell command...
Writing script.rc...
Executing metasploit command...
```

31. Once the script finishes, open the file (**script.rc**) in any text editor. The file can be found in **/home/pentester/firmadyne** directory



```
setg RHOST 192.168.0.100
setg RHOSTS 192.168.0.100

spool exploits/exploit.0.log
use exploits/linux/http/airties_login_cgi_bof
exploit -z
spool off
sessions -K

spool exploits/exploit.1.log
use exploits/linux/http/belkin_login_bof
exploit -z
spool off
sessions -K

spool exploits/exploit.2.log
use exploits/linux/http/ddwrt_cgibin_exec
exploit -z
spool off
sessions -K

spool exploits/exploit.3.log
use exploits/linux/http/dlink_authentication_cgi_bof
exploit -z
spool off
sessions -K
```

32. As you can see in the above screenshot, the tool is only trying a list of exploits. Therefore, it is not a powerful and effective script. Therefore, it should be used as a reference. Similar to other findings, your ability to research will determine success. The exercise objectives have been achieved.