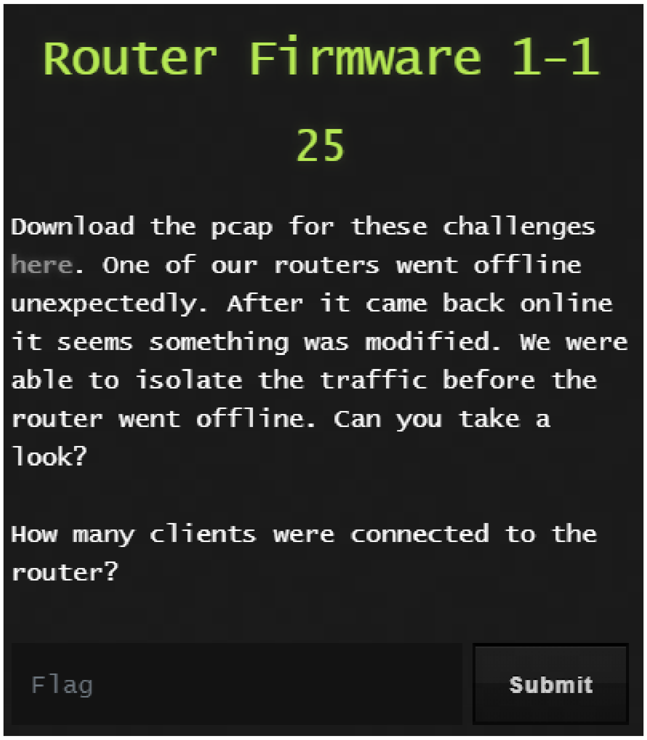


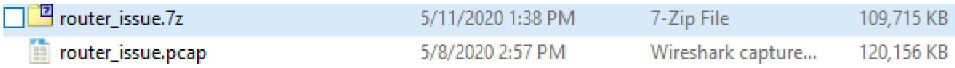
Router Firmware 1-1

25

Download the pcap for these challenges here. One of our routers went offline unexpectedly. After it came back online it seems something was modified. We were able to isolate the traffic before the router went offline. Can you take a look?

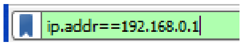
How many clients were connected to the router?

Download and decompress the zip file.



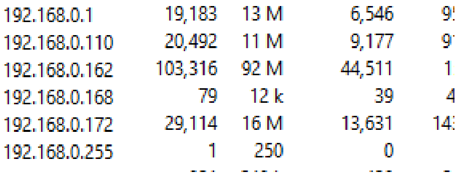
router_issue.7z	5/11/2020 1:38 PM	7-Zip File	109,715 KB
router_issue.pcap	5/8/2020 2:57 PM	Wireshark capture...	120,156 KB

Filter on the router which is at ip.addr==192.168.0.1



```
ip.addr==192.168.0.1
```

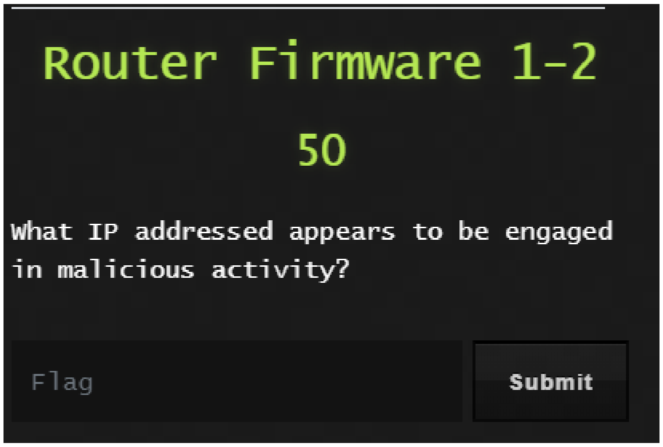
Then going to endpoints, we see that there are 4 devices in the 192.168.0.0/24 subnet.



192.168.0.1	19,183	13 M	6,546	9
192.168.0.110	20,492	11 M	9,177	9
192.168.0.162	103,316	92 M	44,511	1
192.168.0.168	79	12 k	39	4
192.168.0.172	29,114	16 M	13,631	14
192.168.0.255	1	250	0	

1 is the router and 255 is a broadcast.

Flag = 4

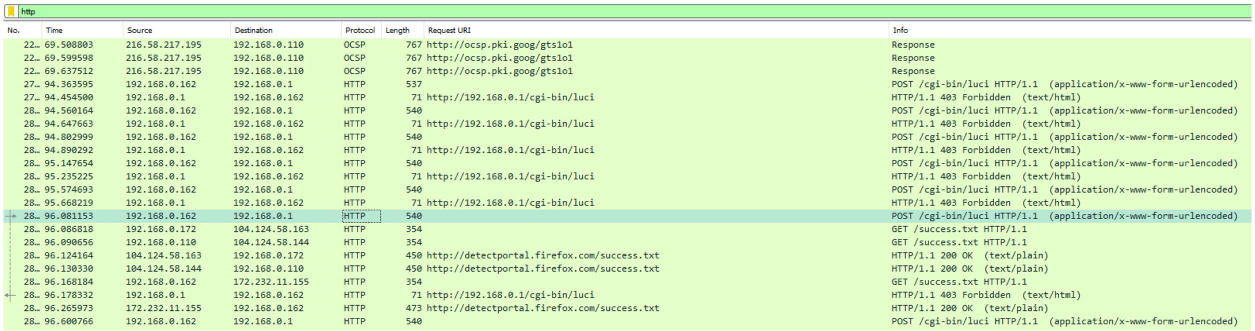


Router Firmware 1-2

50

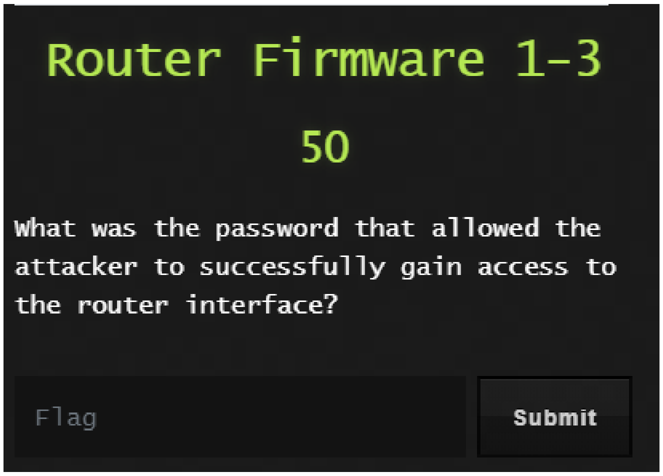
What IP addressed appears to be engaged in malicious activity?

Looking at the data if we look at http traffic, we can see that there is a client accessing the router though a web browser and getting 403 errors trying to POST data.



No.	Time	Source	Destination	Protocol	Length	Request URI	Info
22…	69.508803	216.58.217.195	192.168.0.110	OCSP	767	http://ocsp.pki.goog/gts1o1	Response
22…	69.599598	216.58.217.195	192.168.0.110	OCSP	767	http://ocsp.pki.goog/gts1o1	Response
22…	69.637512	216.58.217.195	192.168.0.110	OCSP	767	http://ocsp.pki.goog/gts1o1	Response
27…	94.363595	192.168.0.162	192.168.0.1	HTTP	537		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
27…	94.454500	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	94.560164	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
28…	94.647663	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	94.802999	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
28…	94.890292	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	95.147654	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
28…	95.235225	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	95.574693	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
28…	95.668219	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	96.081153	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)
28…	96.086818	192.168.0.172	104.124.58.163	HTTP	354		GET /success.txt HTTP/1.1
28…	96.090656	192.168.0.110	104.124.58.144	HTTP	354		GET /success.txt HTTP/1.1
28…	96.124164	104.124.58.163	192.168.0.172	HTTP	450	http://detectportal.firefox.com/success.txt	HTTP/1.1 200 OK (text/plain)
28…	96.130330	104.124.58.144	192.168.0.110	HTTP	450	http://detectportal.firefox.com/success.txt	HTTP/1.1 200 OK (text/plain)
28…	96.168184	192.168.0.162	172.232.11.155	HTTP	354		GET /success.txt HTTP/1.1
28…	96.178332	192.168.0.1	192.168.0.162	HTTP	71	http://192.168.0.1/cgi-bin/luci	HTTP/1.1 403 Forbidden (text/html)
28…	96.265973	172.232.11.155	192.168.0.162	HTTP	473	http://detectportal.firefox.com/success.txt	HTTP/1.1 200 OK (text/plain)
28…	96.600766	192.168.0.162	192.168.0.1	HTTP	540		POST /cgi-bin/luci HTTP/1.1 (application/x-www-form-urlencoded)

Flag: 192.168.0.162



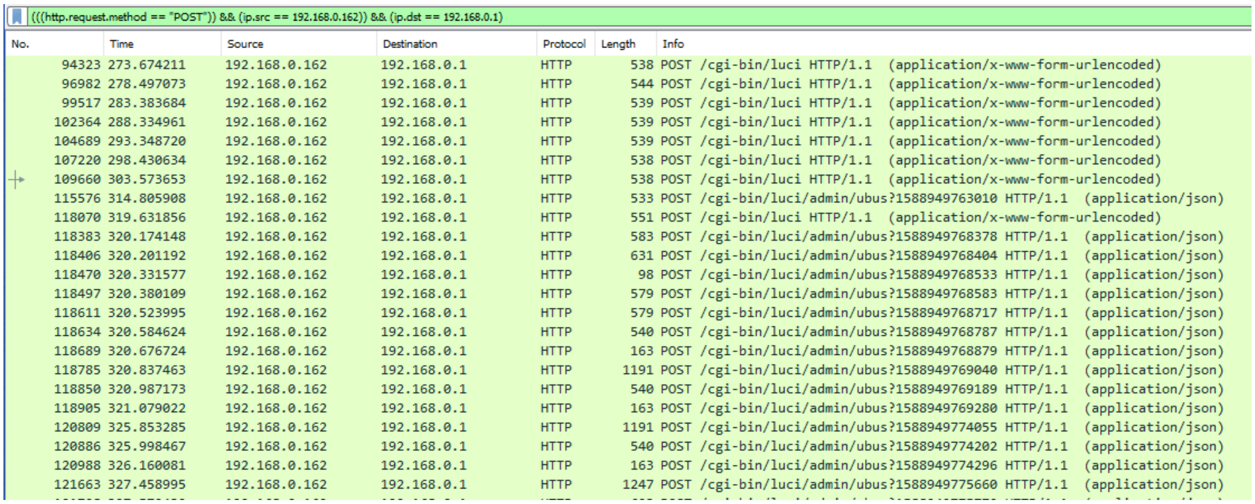
Router Firmware 1-3

50

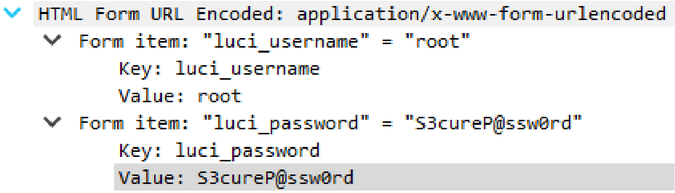
What was the password that allowed the attacker to successfully gain access to the router interface?

Let’s use the following filter to get the traffic between the router and the 162 devices with only the POST data: (((http.request.method == "POST")) && (ip.src == 192.168.0.162)) && (ip.dst == 192.168.0.1)

We can see that there was an attempt that was successful and if we go to packet 118070, we see the username and password used to login.

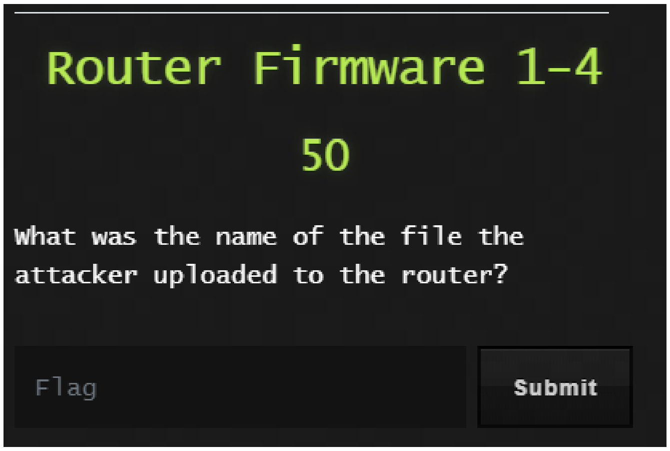


```
(((http.request.method == "POST")) && (ip.src == 192.168.0.162)) && (ip.dst == 192.168.0.1)

No.       Time        Source          Destination    Protocol  Length  Info
   94323 273.674211  192.168.0.162   192.168.0.1    HTTP       538 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
   96982 278.497073  192.168.0.162   192.168.0.1    HTTP       544 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
   99517 283.383684  192.168.0.162   192.168.0.1    HTTP       539 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  102364 288.334961  192.168.0.162   192.168.0.1    HTTP       539 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  104689 293.348720  192.168.0.162   192.168.0.1    HTTP       539 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  107220 298.430634  192.168.0.162   192.168.0.1    HTTP       538 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  109660 303.573653  192.168.0.162   192.168.0.1    HTTP       538 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  115576 314.805908  192.168.0.162   192.168.0.1    HTTP       533 POST /cgi-bin/luci/admin/ubus?1588949763010 HTTP/1.1  (application/json)
  118070 319.631856  192.168.0.162   192.168.0.1    HTTP       551 POST /cgi-bin/luci HTTP/1.1  (application/x-www-form-urlencoded)
  118383 320.174148  192.168.0.162   192.168.0.1    HTTP       583 POST /cgi-bin/luci/admin/ubus?1588949768378 HTTP/1.1  (application/json)
  118406 320.201192  192.168.0.162   192.168.0.1    HTTP       631 POST /cgi-bin/luci/admin/ubus?1588949768404 HTTP/1.1  (application/json)
  118470 320.331577  192.168.0.162   192.168.0.1    HTTP        98 POST /cgi-bin/luci/admin/ubus?1588949768533 HTTP/1.1  (application/json)
  118497 320.380109  192.168.0.162   192.168.0.1    HTTP       579 POST /cgi-bin/luci/admin/ubus?1588949768583 HTTP/1.1  (application/json)
  118611 320.523995  192.168.0.162   192.168.0.1    HTTP       579 POST /cgi-bin/luci/admin/ubus?1588949768717 HTTP/1.1  (application/json)
  118634 320.584624  192.168.0.162   192.168.0.1    HTTP       540 POST /cgi-bin/luci/admin/ubus?1588949768787 HTTP/1.1  (application/json)
  118689 320.676724  192.168.0.162   192.168.0.1    HTTP       163 POST /cgi-bin/luci/admin/ubus?1588949768879 HTTP/1.1  (application/json)
  118785 320.837463  192.168.0.162   192.168.0.1    HTTP      1191 POST /cgi-bin/luci/admin/ubus?1588949769040 HTTP/1.1  (application/json)
  118850 320.987173  192.168.0.162   192.168.0.1    HTTP       540 POST /cgi-bin/luci/admin/ubus?1588949769189 HTTP/1.1  (application/json)
  118905 321.079022  192.168.0.162   192.168.0.1    HTTP       163 POST /cgi-bin/luci/admin/ubus?1588949769280 HTTP/1.1  (application/json)
  120809 325.853285  192.168.0.162   192.168.0.1    HTTP      1191 POST /cgi-bin/luci/admin/ubus?1588949774055 HTTP/1.1  (application/json)
  120886 325.998467  192.168.0.162   192.168.0.1    HTTP       540 POST /cgi-bin/luci/admin/ubus?1588949774202 HTTP/1.1  (application/json)
  120988 326.160081  192.168.0.162   192.168.0.1    HTTP       163 POST /cgi-bin/luci/admin/ubus?1588949774296 HTTP/1.1  (application/json)
  121663 327.458995  192.168.0.162   192.168.0.1    HTTP      1247 POST /cgi-bin/luci/admin/ubus?1588949775660 HTTP/1.1  (application/json)
```



```
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "luci_username" = "root"
      Key: luci_username
      Value: root
  Form item: "luci_password" = "S3cureP@ssw0rd"
      Key: luci_password
      Value: S3cureP@ssw0rd
```

Flag: S3cureP@ssw0rd

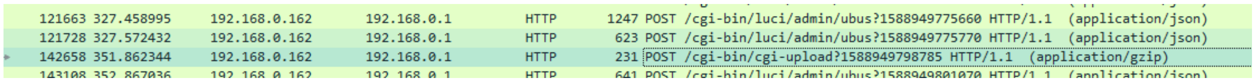


Router Firmware 1-4

50

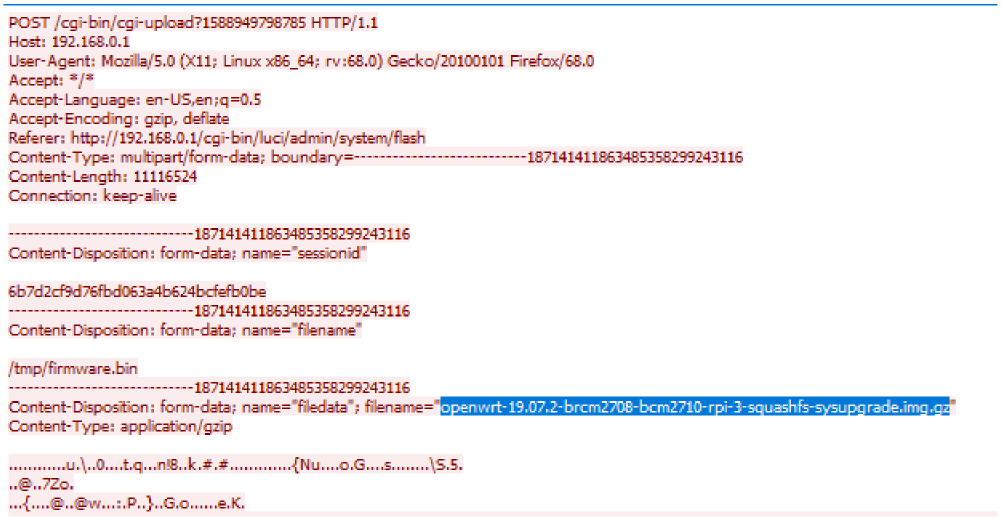
What was the name of the file the attacker uploaded to the router?

If we continue further down, we see that there was an upload to the router.



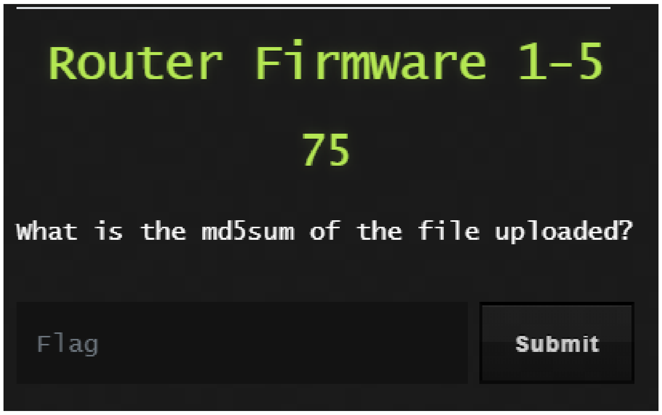
```
121663 327.458995   192.168.0.162   192.168.0.1   HTTP   1247 POST /cgi-bin/luci/admin/ubus?1588949775660 HTTP/1.1  (application/json)
121728 327.572432   192.168.0.162   192.168.0.1   HTTP    623 POST /cgi-bin/luci/admin/ubus?1588949775770 HTTP/1.1  (application/json)
142658 351.862344   192.168.0.162   192.168.0.1   HTTP    231 POST /cgi-bin/cgi-upload?1588949798785 HTTP/1.1  (application/gzip)
143108 352.867036   192.168.0.162   192.168.0.1   HTTP    641 POST /cgi-bin/luci/admin/ubus?1588949801070 HTTP/1.1  (application/json)
```

Follow the https Stream of the data to get the filename.



```
POST /cgi-bin/cgi-upload?1588949798785 HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.1/cgi-bin/luci/admin/system/flash
Content-Type: multipart/form-data; boundary=---------------------------187141411863485358299243116
Content-Length: 11116524
Connection: keep-alive

-----------------------------187141411863485358299243116
Content-Disposition: form-data; name="sessionid"

6b7d2cf9d76fbd063a4b624bcfefb0be
-----------------------------187141411863485358299243116
Content-Disposition: form-data; name="filename"

/tmp/firmware.bin
-----------------------------187141411863485358299243116
Content-Disposition: form-data; name="filedata"; filename="openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.gz"
Content-Type: application/gzip

...........u.\..0....t.q...n!8..k.#.#............{Nu....o.G....s........\S.5.
..@..7Zo.
...{....@..@w...:.P..}..G.o......e.K.
```

Flag: openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.gz

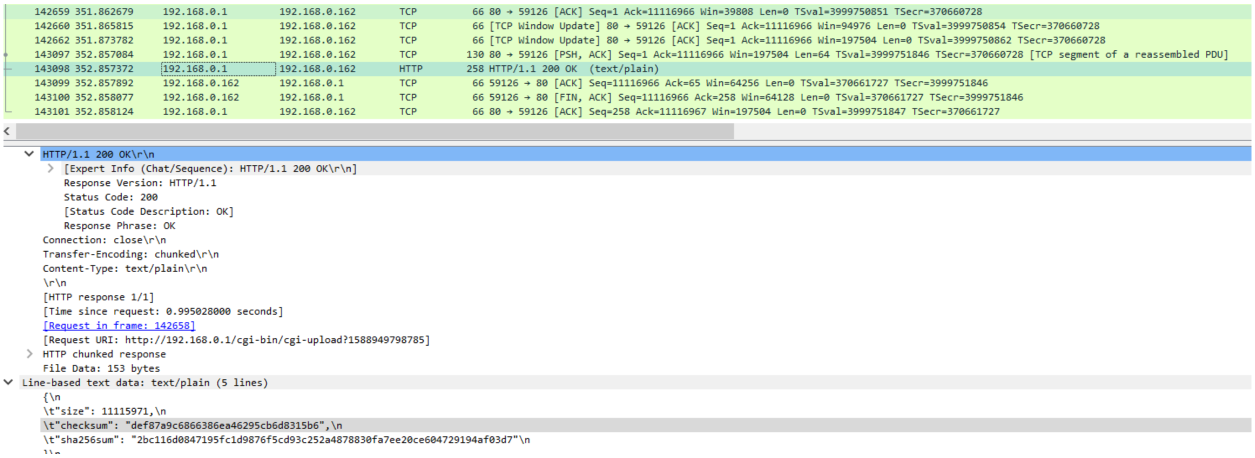


Router Firmware 1-5

75

What is the md5sum of the file uploaded?

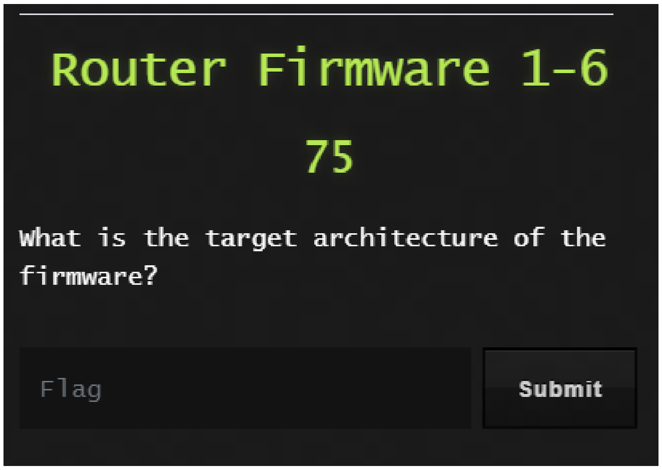
In the stream that we follow we have the uploaded file and a text file. The text file contains that checksum of the file that was uploaded.



```
142659 351.862679   192.168.0.1     192.168.0.162   TCP     66 80 → 59126 [ACK] Seq=1 Ack=11116966 Win=39808 Len=0 TSval=3999750851 TSecr=370660728
142660 351.865815   192.168.0.1     192.168.0.162   TCP     66 [TCP Window Update] 80 → 59126 [ACK] Seq=1 Ack=11116966 Win=94976 Len=0 TSval=3999750854 TSecr=370660728
142662 351.873782   192.168.0.1     192.168.0.162   TCP     66 [TCP Window Update] 80 → 59126 [ACK] Seq=1 Ack=11116966 Win=197504 Len=0 TSval=3999750862 TSecr=370660728
143097 352.857084   192.168.0.1     192.168.0.162   TCP    130 80 → 59126 [PSH, ACK] Seq=1 Ack=11116966 Win=197504 Len=64 TSval=3999751846 TSecr=370660728 [TCP segment of a reassembled PDU]
143098 352.857372   192.168.0.1     192.168.0.162   HTTP   258 HTTP/1.1 200 OK  (text/plain)
143099 352.857892   192.168.0.162   192.168.0.1     TCP     66 59126 → 80 [ACK] Seq=11116966 Ack=65 Win=64256 Len=0 TSval=370661727 TSecr=3999751846
143100 352.858077   192.168.0.162   192.168.0.1     TCP     66 59126 → 80 [FIN, ACK] Seq=11116966 Ack=258 Win=64128 Len=0 TSval=370661727 TSecr=3999751846
143101 352.858124   192.168.0.1     192.168.0.162   TCP     66 80 → 59126 [ACK] Seq=258 Ack=11116967 Win=197504 Len=0 TSval=3999751847 TSecr=370661727
```

```
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Connection: close\r\n
  Transfer-Encoding: chunked\r\n
  Content-Type: text/plain\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.995028000 seconds]
  [Request in frame: 142658]
  [Request URI: http://192.168.0.1/cgi-bin/cgi-upload?1588949798785]
  HTTP chunked response
  File Data: 153 bytes
Line-based text data: text/plain (5 lines)
  {\n
  \t"size": 11115971,\n
  \t"checksum": "def87a9c6866386ea46295cb6d8315b6",\n
  \t"sha256sum": "2bc116d0847195fc1d9876f5cd93c252a4878830fa7ee20ce604729194af03d7"\n
  }\n
```

Flag: def87a9c6866386ea46295cb6d8315b6

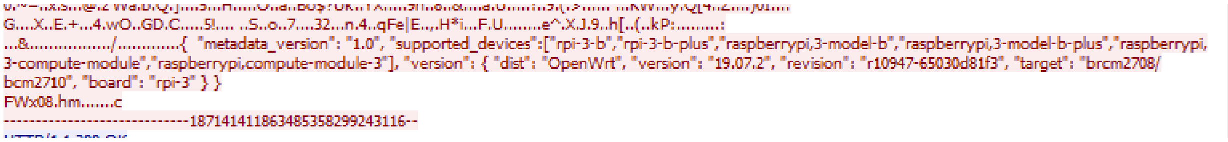


Router Firmware 1-6

75

What is the target architecture of the firmware?

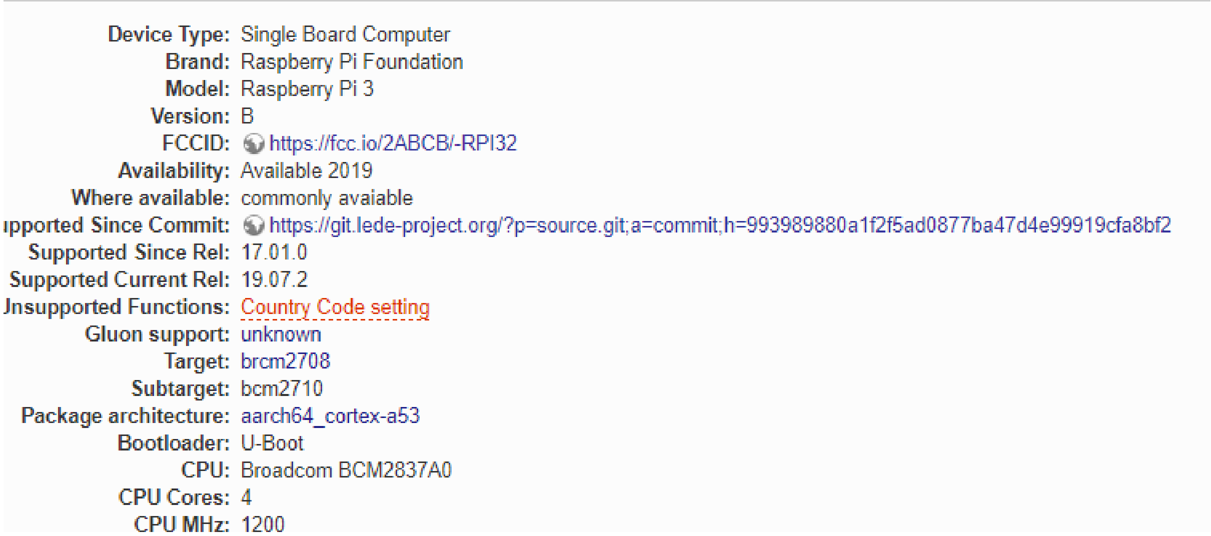
We see that this is going to be looking for a raspberry pi 3:



```
G....X..E.+...4.wO..GD.C.....5!.... ..S..o..7....32...n.4..qFe|E..,.H*i...F.U........e^.X.J.9..h[..(..kP:.........:
...&................./............{  "metadata_version": "1.0", "supported_devices":["rpi-3-b","rpi-3-b-plus","raspberrypi,3-model-b","raspberrypi,3-model-b-plus","raspberrypi,
3-compute-module","raspberrypi,compute-module-3"], "version": { "dist": "OpenWrt", "version": "19.07.2", "revision": "r10947-65030d81f3", "target": "brcm2708/
bcm2710", "board": "rpi-3" } }
FWx08.hm.......c
-----------------------------187141411863485358299243116--
```

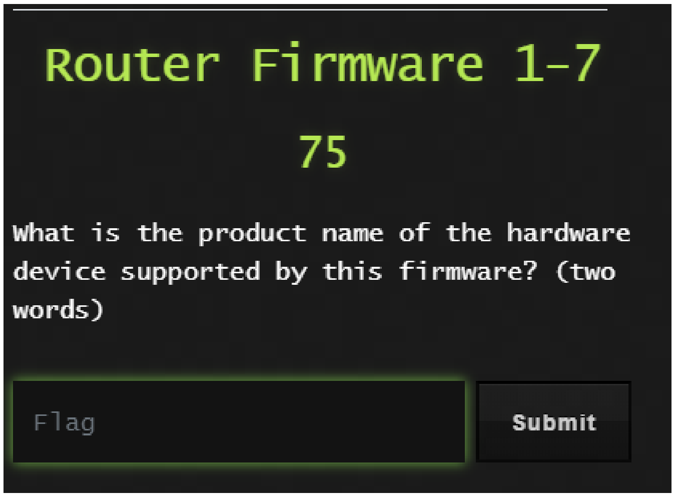
So, we look for wrt and the raspberry pi 3 at the following location to get the architecture.

https://openwrt.org/toh/hwdata/raspberry_pi_foundation/raspberry_pi_3_b



Device Type:	Single Board Computer
Brand:	Raspberry Pi Foundation
Model:	Raspberry Pi 3
Version:	B
FCCID:	https://fcc.io/2ABCB/-RPI32
Availability:	Available 2019
Where available:	commonly avaiable
upported Since Commit:	https://git.lede-project.org/?p=source.git;a=commit;h=993989880a1f2f5ad0877ba47d4e99919cfa8bf2
Supported Since Rel:	17.01.0
Supported Current Rel:	19.07.2
Unsupported Functions:	Country Code setting
Gluon support:	unknown
Target:	brcm2708
Subtarget:	bcm2710
Package architecture:	aarch64_cortex-a53
Bootloader:	U-Boot
CPU:	Broadcom BCM2837A0
CPU Cores:	4
CPU MHz:	1200

Flag: aarch64_cortex-a53



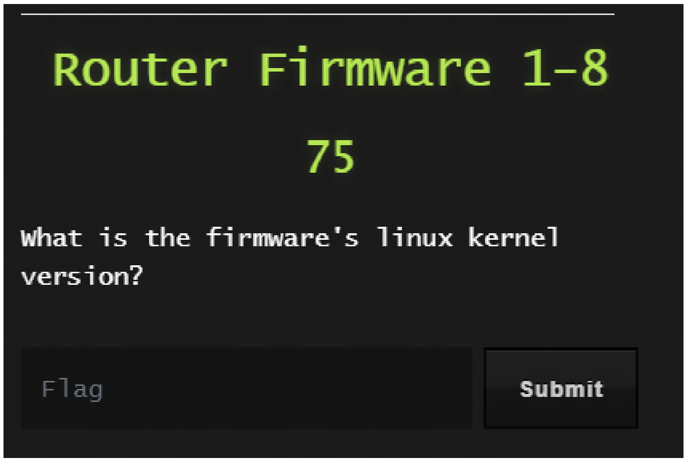
Router Firmware 1-7

75

What is the product name of the hardware device supported by this firmware? (two words)

We saw that this was looking for the raspberry pi 3b platform from that last couple of questions.

Flag: Raspberry Pi

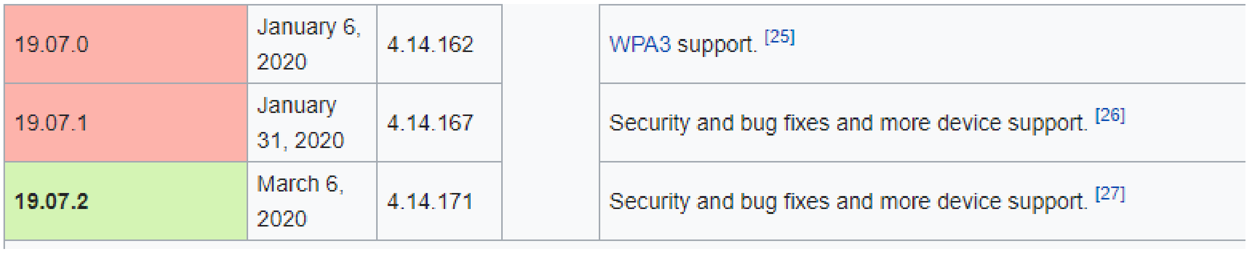


Router Firmware 1-8

75

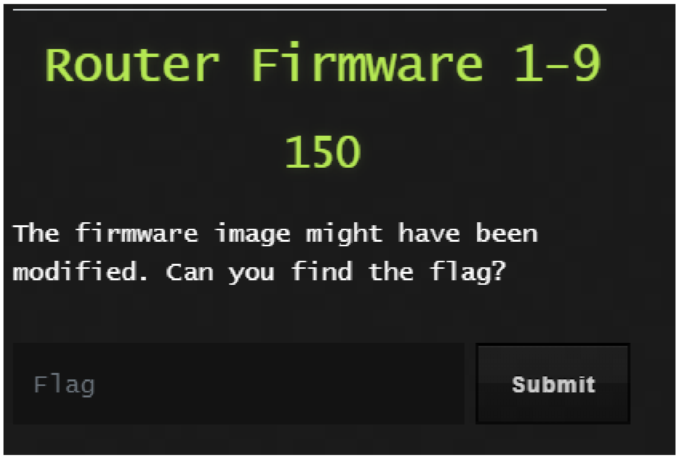
What is the firmware's linux kernel version?

We know that it was loading openwrt 19.07.2 from the file, a quick search on google gives us a link to Wikipedia that lists the underlying kernel version.



19.07.0	January 6, 2020	4.14.162		WPA3 support. [25]
19.07.1	January 31, 2020	4.14.167		Security and bug fixes and more device support. [26]
19.07.2	March 6, 2020	4.14.171		Security and bug fixes and more device support. [27]

Flag: 4.14.171



Router Firmware 1-9

150

The firmware image might have been modified. Can you find the flag?

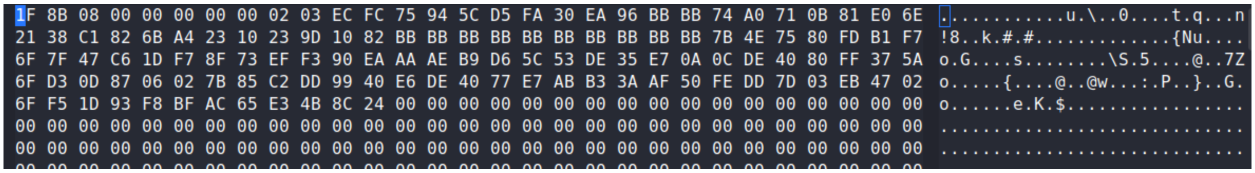
This will take some time to get through, but here is the process that we are going to follow to get the answer. We need to compare the firmware from the pcap to the legitimate firmware, and from there we need to see what is different/new.

To start off we will carve out the pcap file, save the tcp stream 1-6 to a file.

Download the actual firmware from openwrt: https://downloads.openwrt.org/releases/19.07.2/targets/brcm2708/bcm2710/ we want the squashfs version.

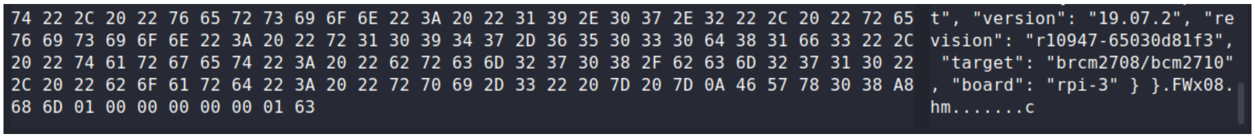
Next we will need to take the pcap file and clip off some hex bits at the front and end. (I used ghex)

Delete everything before the bits 1F 8B 08 00 00 000



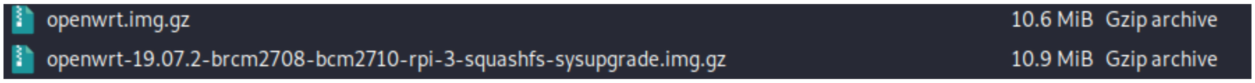
```
1F 8B 08 00 00 00 00 00 02 03 EC FC 75 94 5C D5 FA 30 EA 96 BB BB 74 A0 71 0B 81 E0 6E  ...........u.\..0....t.q...n
21 38 C1 82 6B A4 23 10 23 9D 10 82 BB BB BB BB BB BB BB BB BB BB 7B 4E 75 80 FD B1 F7  !8..k.#.#.............{Nu....
6F 7F 47 C6 1D F7 8F 73 EF F3 90 EA AA AE B9 D6 5C 53 DE 35 E7 0A 0C DE 40 80 FF 37 5A  o.G....s........\S.5....@..7Z
6F D3 0D 87 06 02 7B 85 C2 DD 99 40 E6 DE 40 77 E7 AB B3 3A AF 50 FE DD 7D 03 EB 47 02  o.....{....@..@w...:.P..}..G.
6F F5 1D 93 F8 BF AC 65 E3 4B 8C 24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  o......e.K.$.................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
```

At the end of the file delete everything after 68 6D 01 00 00 00 00 01 63



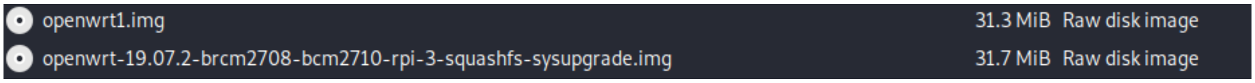
```
74 22 2C 20 22 76 65 72 73 69 6F 6E 22 3A 20 22 31 39 2E 30 37 2E 32 22 2C 20 22 72 65  t", "version": "19.07.2", "re
76 69 73 69 6F 6E 22 3A 20 22 72 31 30 39 34 37 2D 36 35 30 33 30 64 38 31 66 33 22 2C  vision": "r10947-65030d81f3",
20 22 74 61 72 67 65 74 22 3A 20 22 62 72 63 6D 32 37 30 38 2F 62 63 6D 32 37 31 30 22   "target": "brcm2708/bcm2710"
2C 20 22 62 6F 61 72 64 22 3A 20 22 72 70 69 2D 33 22 20 7D 20 7D 0A 46 57 78 30 38 A8  , "board": "rpi-3" } }.FWx08.
68 6D 01 00 00 00 00 00 01 63                                                           hm.......c
```

Now we have two img.gz files that should be the exact same (other than what was modified).



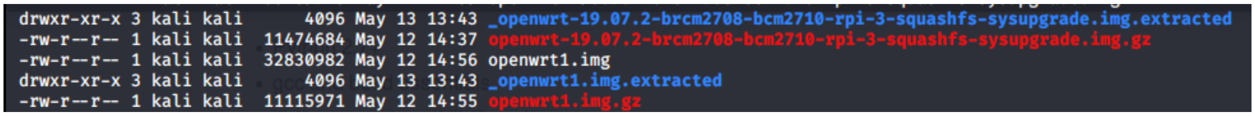
```
openwrt.img.gz                                                    10.6 MiB  Gzip archive
openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.gz 10.9 MiB  Gzip archive
```

Unzip the files to get the img files.



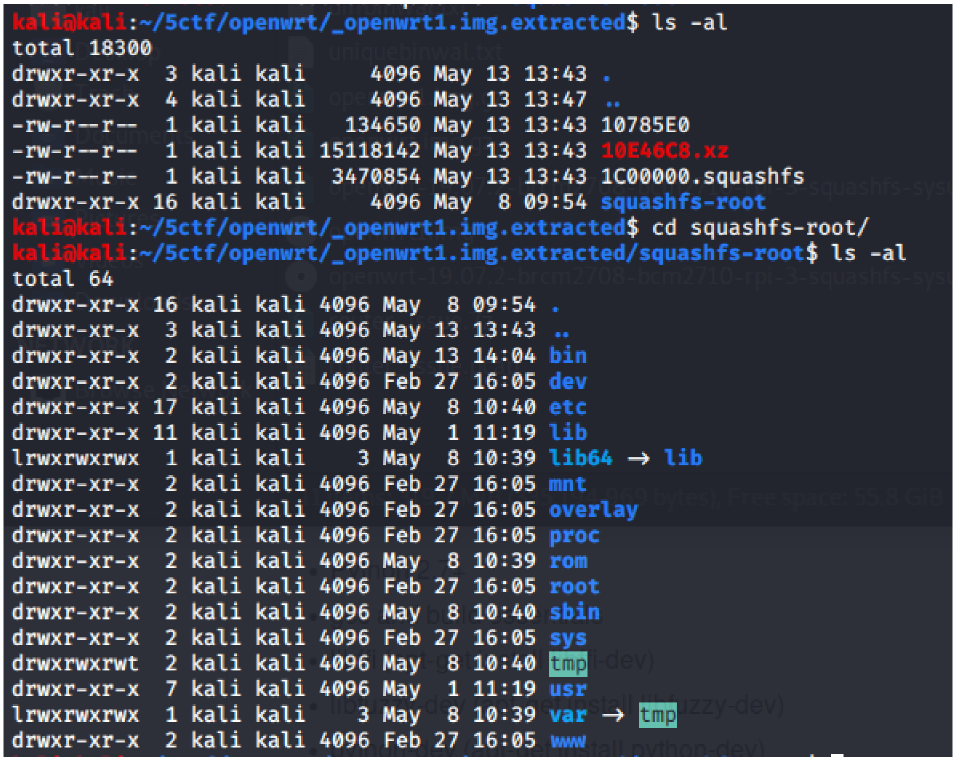
```
openwrt1.img                                                   31.3 MiB  Raw disk image
openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img 31.7 MiB  Raw disk image
```

Next we will do a binwalk -e on both files to dump the directories to browse.



```
drwxr-xr-x 3 kali kali     4096 May 13 13:43 _openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted
-rw-r--r-- 1 kali kali 11474684 May 12 14:37 openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.gz
-rw-r--r-- 1 kali kali 32830982 May 12 14:56 openwrt1.img
drwxr-xr-x 3 kali kali     4096 May 13 13:43 _openwrt1.img.extracted
-rw-r--r-- 1 kali kali 11115971 May 12 14:55 openwrt1.img.gz
```

We can look at the filesystem now



```
kali@kali:~/5ctf/openwrt/_openwrt1.img.extracted$ ls -al
total 18300
drwxr-xr-x  3 kali kali     4096 May 13 13:43 .
drwxr-xr-x  4 kali kali     4096 May 13 13:47 ..
-rw-r--r--  1 kali kali   134650 May 13 13:43 10785E0
-rw-r--r--  1 kali kali 15118142 May 13 13:43 10E46C8.xz
-rw-r--r--  1 kali kali  3470854 May 13 13:43 1C00000.squashfs
drwxr-xr-x 16 kali kali     4096 May  8 09:54 squashfs-root
kali@kali:~/5ctf/openwrt/_openwrt1.img.extracted$ cd squashfs-root/
kali@kali:~/5ctf/openwrt/_openwrt1.img.extracted/squashfs-root$ ls -al
total 64
drwxr-xr-x 16 kali kali 4096 May  8 09:54 .
drwxr-xr-x  3 kali kali 4096 May 13 13:43 ..
drwxr-xr-x  2 kali kali 4096 May 13 14:04 bin
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 dev
drwxr-xr-x 17 kali kali 4096 May  8 10:40 etc
drwxr-xr-x 11 kali kali 4096 May  1 11:19 lib
lrwxrwxrwx  1 kali kali    3 May  8 10:39 lib64 → lib
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 mnt
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 overlay
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 proc
drwxr-xr-x  2 kali kali 4096 May  8 10:39 rom
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 root
drwxr-xr-x  2 kali kali 4096 May  8 10:40 sbin
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 sys
drwxrwxrwt  2 kali kali 4096 May  8 10:40 tmp
drwxr-xr-x  7 kali kali 4096 May  1 11:19 usr
lrwxrwxrwx  1 kali kali    3 May  8 10:39 var → tmp
drwxr-xr-x  2 kali kali 4096 Feb 27 16:05 www
```

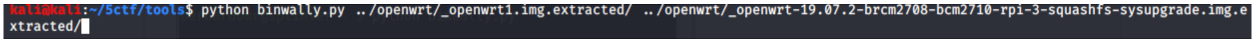
next we need to install binwally.py as this is a neat tool that will show the differences between the two directories. Make sure to install the prereqs.

https://github.com/bmaia/binwally/blob/master/README.md

#Prerequisites:

- Python 2.7+
- gcc and build essentials
- libffi (apt-get install libffi-dev)
- libfuzzy-dev (apt-get install libfuzzy-dev)
- python-dev (apt-get install python-dev)
- python-ssdeep (pip install ssdeep)

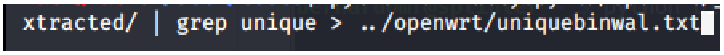
To run this command, it is as simple as python binwally.py dir1 dir2



```
kali@kali:~/5ctf/tools$ python binwally.py ../openwrt/_openwrt1.img.extracted/ ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.e
xtracted/
```

this will compare the two binwalk extracted directories

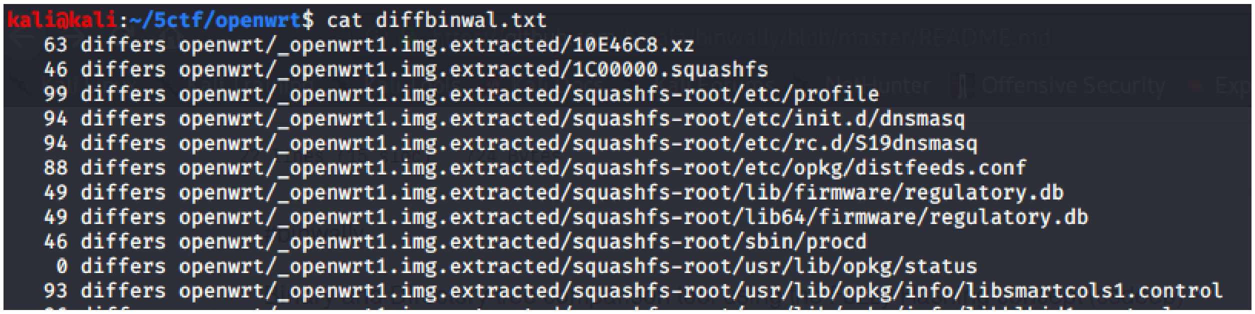
I added a term at the end to grep for all unique lines and output this to a file to look at



```
xtracted/ | grep unique > ../openwrt/uniquebinwal.txt
```

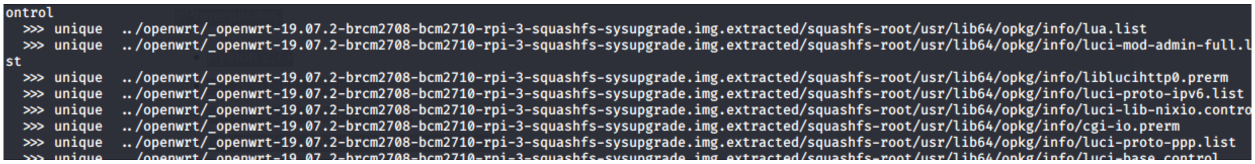
I also did the same the same with differs also if I needed it. Binwally has matches, differs, unique.

Differs file:



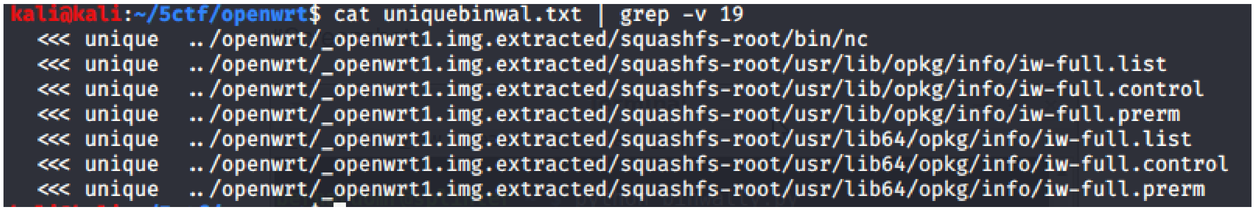
```
kali@kali:~/5ctf/openwrt$ cat diffbinwal.txt
   63 differs openwrt/_openwrt1.img.extracted/10E46C8.xz
   46 differs openwrt/_openwrt1.img.extracted/1C00000.squashfs
   99 differs openwrt/_openwrt1.img.extracted/squashfs-root/etc/profile
   94 differs openwrt/_openwrt1.img.extracted/squashfs-root/etc/init.d/dnsmasq
   94 differs openwrt/_openwrt1.img.extracted/squashfs-root/etc/rc.d/S19dnsmasq
   88 differs openwrt/_openwrt1.img.extracted/squashfs-root/etc/opkg/distfeeds.conf
   49 differs openwrt/_openwrt1.img.extracted/squashfs-root/lib/firmware/regulatory.db
   49 differs openwrt/_openwrt1.img.extracted/squashfs-root/lib64/firmware/regulatory.db
   46 differs openwrt/_openwrt1.img.extracted/squashfs-root/sbin/procd
    0 differs openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib/opkg/status
   93 differs openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib/opkg/info/libsmartcols1.control
```

Unique file:



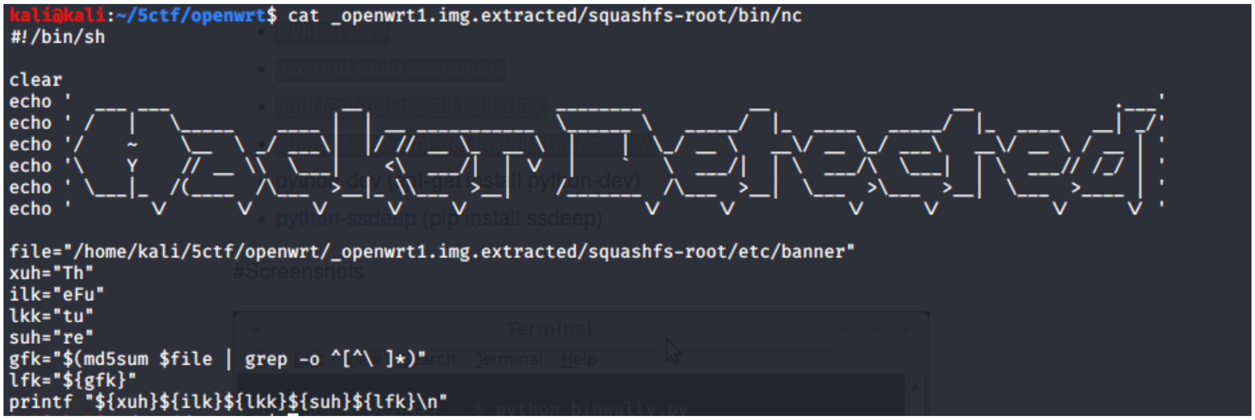
```
ontrol
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/lua.list
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/luci-mod-admin-full.l
st
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/liblucihttp0.prerm
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/luci-proto-ipv6.list
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/luci-lib-nixio.contro
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/cgi-io.prerm
  >>> unique   ../openwrt/_openwrt-19.07.2-brcm2708-bcm2710-rpi-3-squashfs-sysupgrade.img.extracted/squashfs-root/usr/lib64/opkg/info/luci-proto-ppp.list
```

In the unique file we want to filter out the references to the correct file:



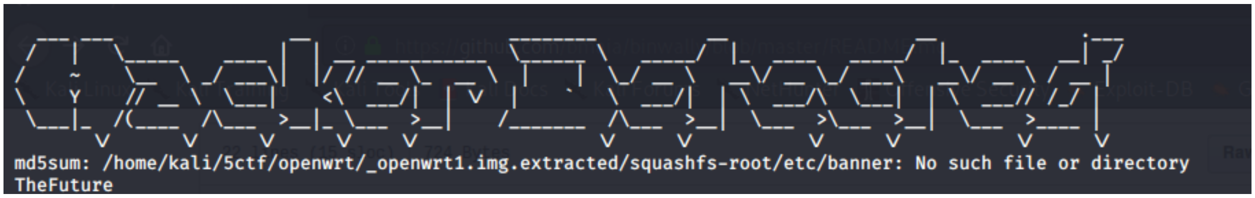
```
kali@kali:~/5ctf/openwrt$ cat uniquebinwal.txt | grep -v 19
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/bin/nc
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib/opkg/info/iw-full.list
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib/opkg/info/iw-full.control
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib/opkg/info/iw-full.prerm
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib64/opkg/info/iw-full.list
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib64/opkg/info/iw-full.control
  <<< unique   ../openwrt/_openwrt1.img.extracted/squashfs-root/usr/lib64/opkg/info/iw-full.prerm
```

We notice that there is a netcat file in the image, when we cat that file it is interesting:



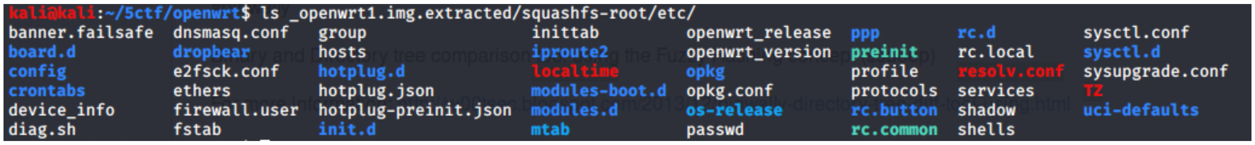
```
kali@kali:~/5ctf/openwrt$ cat _openwrt1.img.extracted/squashfs-root/bin/nc
#!/bin/sh

clear
echo '  ___ ___              __                ________          __            __          .___.'
echo ' /   |   \_____    ____|  | __ ___________ \______ \   _____/  |_  ____   _____/  |_  ____   __| _/'
echo '/    ~    \__  \ _/ ___\  |/ // __ \_  __ \ |    |  \_/ __ \   __\/ __ \_/ ___\   __\/ __ \ / __ | '
echo '\    Y    // __ \\  \___|    <\  ___/|  | \/ |    `   \  ___/|  | \  ___/\  \___|  | \  ___// /_/ | '
echo ' \___|_  /(____  /\___  >__|_ \\___  >__|   /_______  /\___  >__|  \___  >\___  >__|  \___  >____ | '
echo '       \/      \/     \/     \/    \/               \/     \/          \/     \/          \/     \/ '

file="/home/kali/5ctf/openwrt/_openwrt1.img.extracted/squashfs-root/etc/banner"
xuh="Th"
ilk="eFu"
lkk="tu"
suh="re"
gfk="$(md5sum $file | grep -o ^[^\ ]*)"
lfk="${gfk}"
printf "${xuh}${ilk}${lkk}${suh}${lfk}\n"
```

If we run the file, we almost get the flag we are looking for:



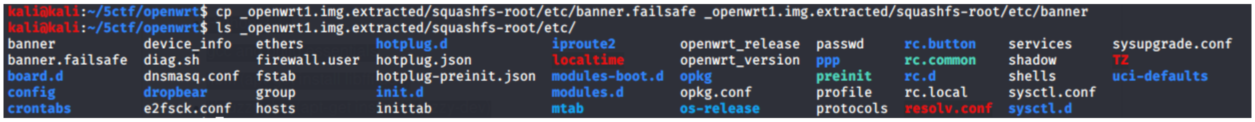
```
md5sum: /home/kali/5ctf/openwrt/_openwrt1.img.extracted/squashfs-root/etc/banner: No such file or directory
TheFuture
```

We need to find the /etc/banner file as it is missing from the uploaded image



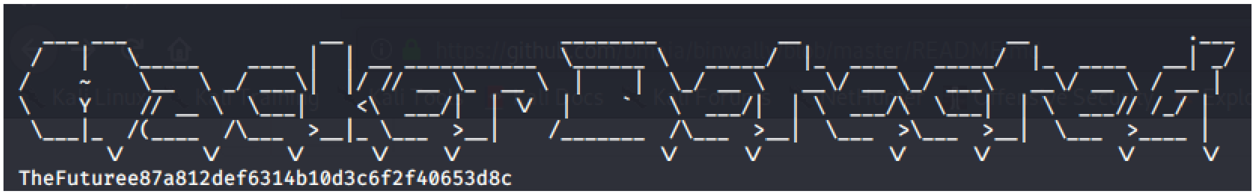
```
kali@kali:~/5ctf/openwrt$ ls _openwrt1.img.extracted/squashfs-root/etc/
banner.failsafe  dnsmasq.conf   group                 inittab          openwrt_release  ppp        rc.d      sysctl.conf
board.d          dropbear       hosts                 iproute2         openwrt_version  preinit    rc.local  sysctl.d
config           e2fsck.conf    hotplug.d             localtime        opkg             profile    resolv.conf  sysupgrade.conf
crontabs         ethers         hotplug.json          modules-boot.d   opkg.conf        protocols  services  TZ
device_info      firewall.user  hotplug-preinit.json  modules.d        os-release       rc.button  shadow    uci-defaults
diag.sh          fstab          init.d                mtab             passwd           rc.common  shells
```

We will copy the failsafe banner as the new banner



```
kali@kali:~/5ctf/openwrt$ cp _openwrt1.img.extracted/squashfs-root/etc/banner.failsafe _openwrt1.img.extracted/squashfs-root/etc/banner
kali@kali:~/5ctf/openwrt$ ls _openwrt1.img.extracted/squashfs-root/etc/
banner           device_info   ethers         hotplug.d             iproute2        openwrt_release  passwd     rc.button    services     sysupgrade.conf
banner.failsafe  diag.sh       firewall.user  hotplug.json          localtime       openwrt_version  ppp        rc.common    shadow       TZ
board.d          dnsmasq.conf  fstab          hotplug-preinit.json  modules-boot.d  opkg             preinit    rc.d         shells       uci-defaults
config           dropbear      group          init.d                modules.d       opkg.conf        profile    rc.local     sysctl.conf
crontabs         e2fsck.conf   hosts          inittab               mtab            os-release       protocols  resolv.conf  sysctl.d
```

And we rerun the nc file and we now have the flag.



```
TheFuturee87a812def6314b10d3c6f2f40653d8c
```

Flag: TheFuturee87a812def6314b10d3c6f2f40653d8c