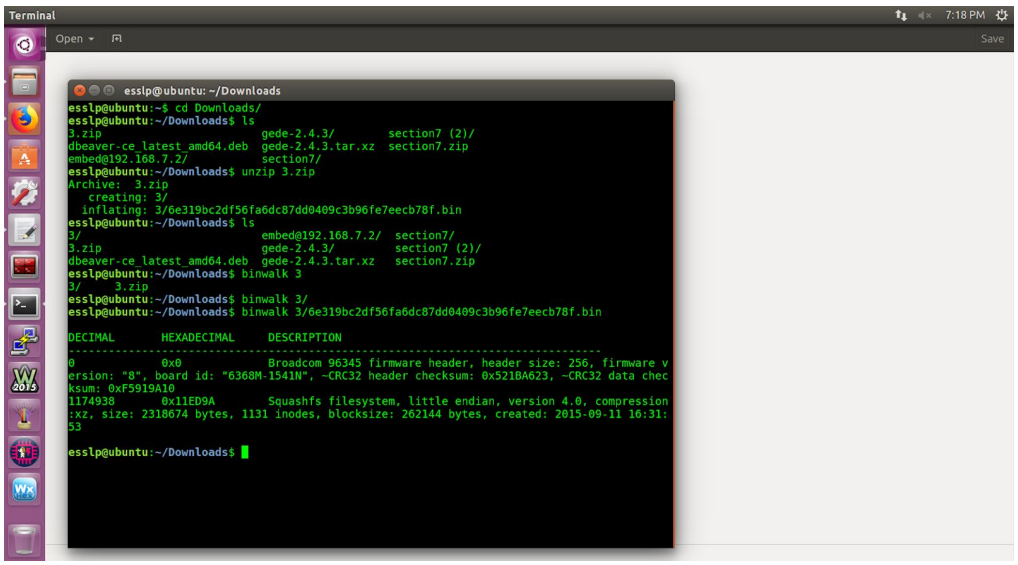
Please submit your answers to the following in a report in pdf format by the deadline given above:

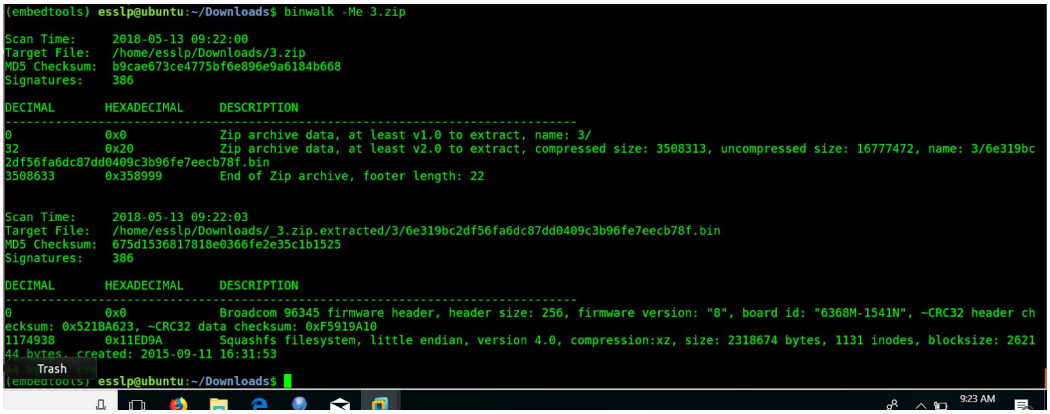
1. Your name: **Khushali Dalal**

2. The name of the firmware file assigned to you. **3.zip**

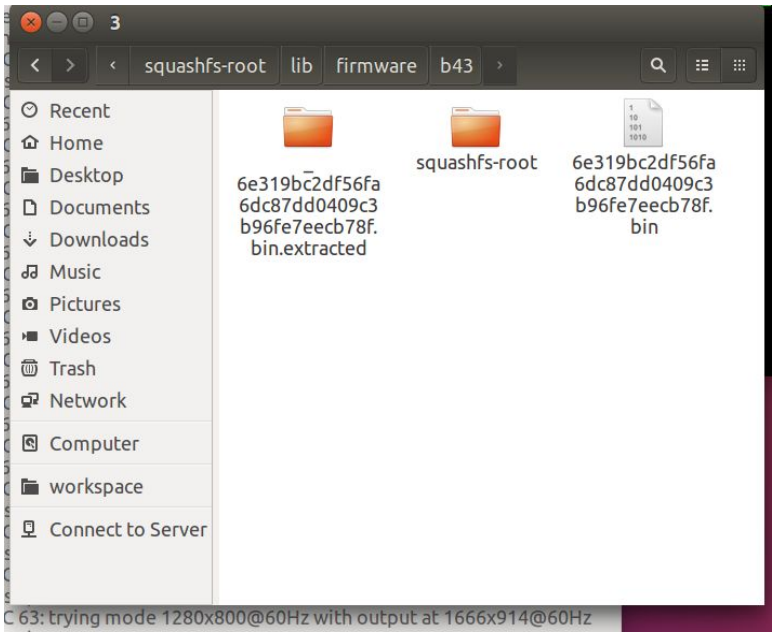
3. List all the contents including files and directories of your firmware up to the system file level and provide a brief high-level description for each file, similar to the example shown below. Also explain how you extracted the firmware contents ((whether it is by using binwalk or manually). Note that some firmware may have multiple firmware(s) embedded within them.

After unzipping the given firmware file, I used Binwalk to extract the files from .bin file. Below is the screenshot of the files extracted after running command
binwalk <file name>

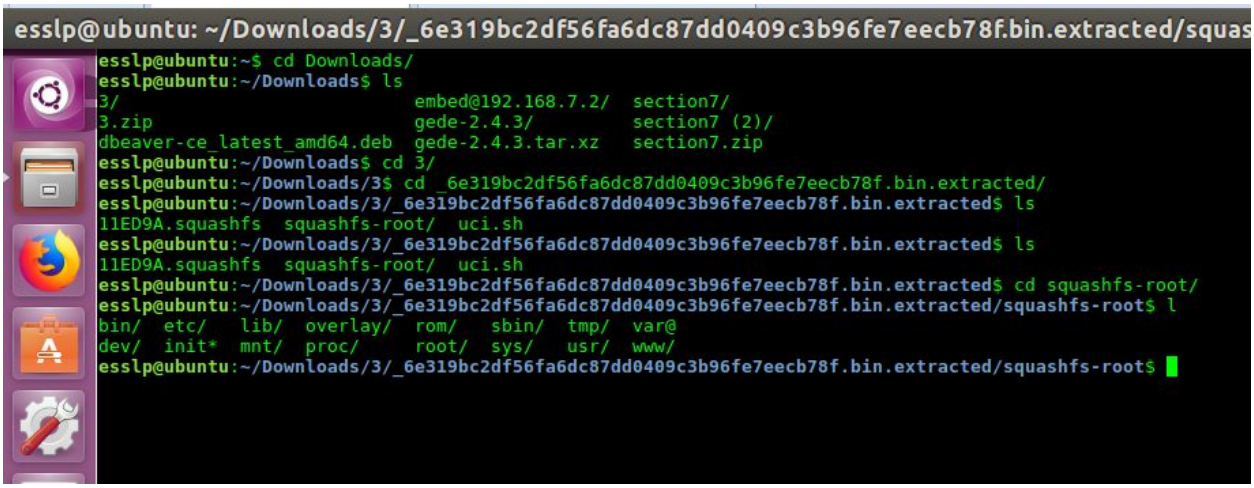




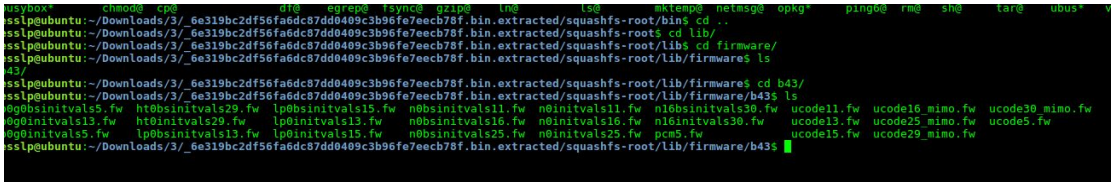
These were the extracted files using binwalk. The version number of firmware is ‘8’. It is little endian. Other useful information are board id, checksum and creation dat

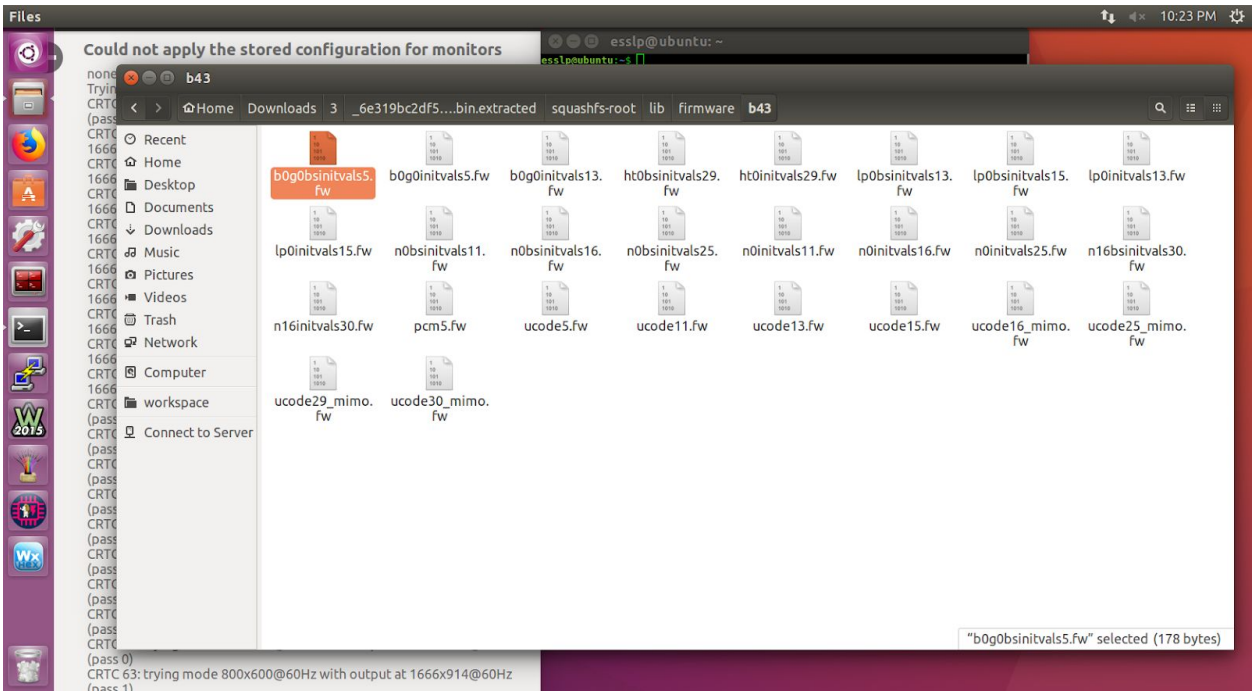


In the end these were the system level files found in squashfs-root folder extracted from the given firmware.



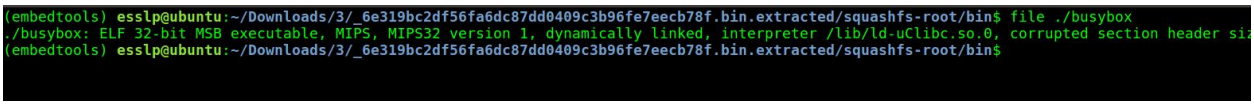
Total firmwares found in the lib/firmware folder are as shown in below screenshot:



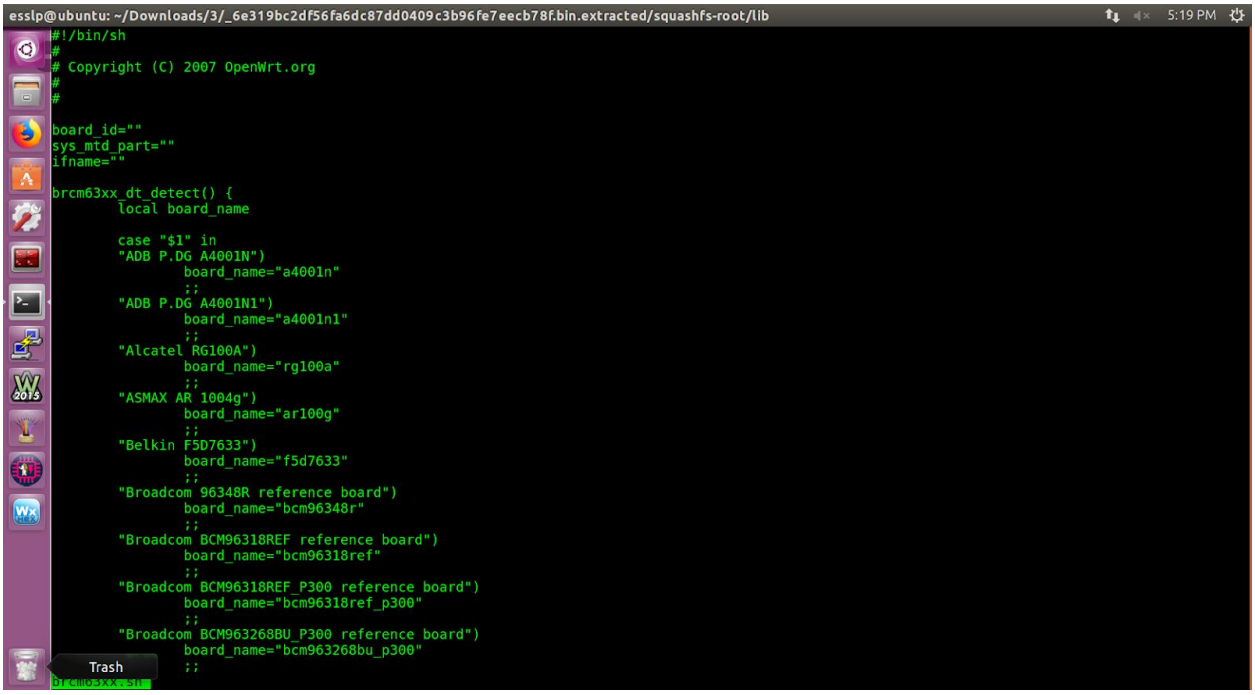


These were some of the the broad names found in file brcm63xx.sh in lib folder.

To check endianness and architecture details I tried file utility on busybox found in bin directory. I.e file ./busybox ; the output is as shown below



Hence, this firmware is designed for big endian (MSB) MIPS architecture.

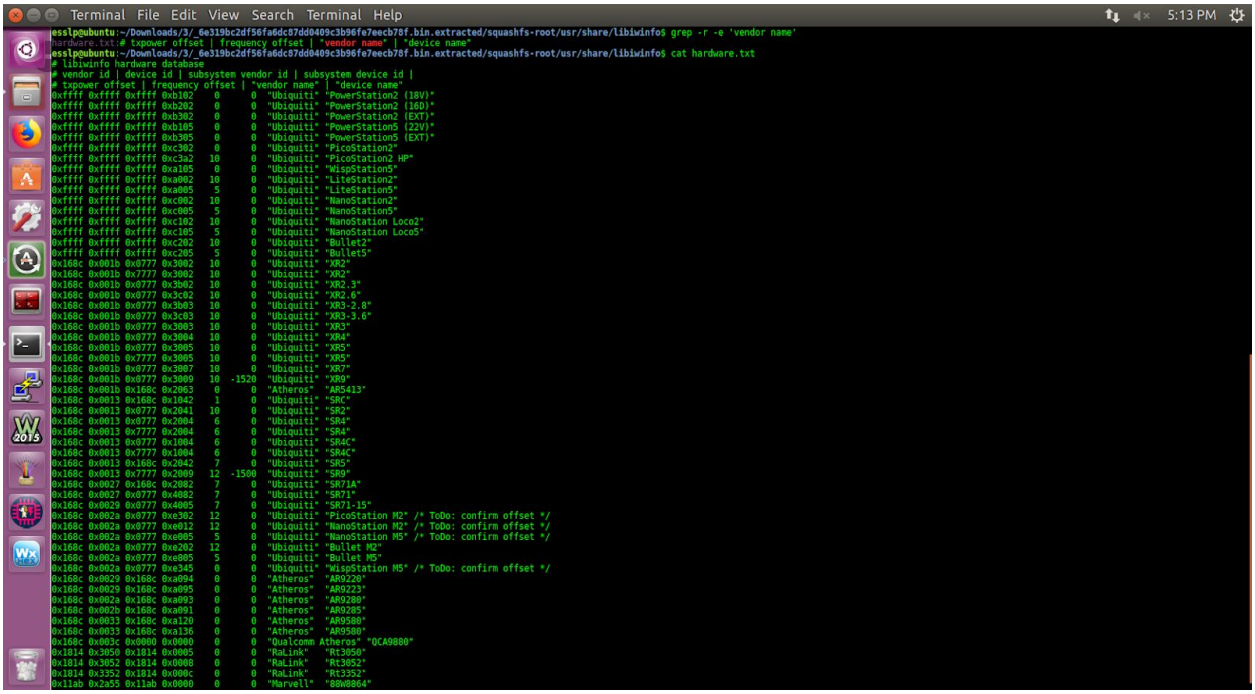


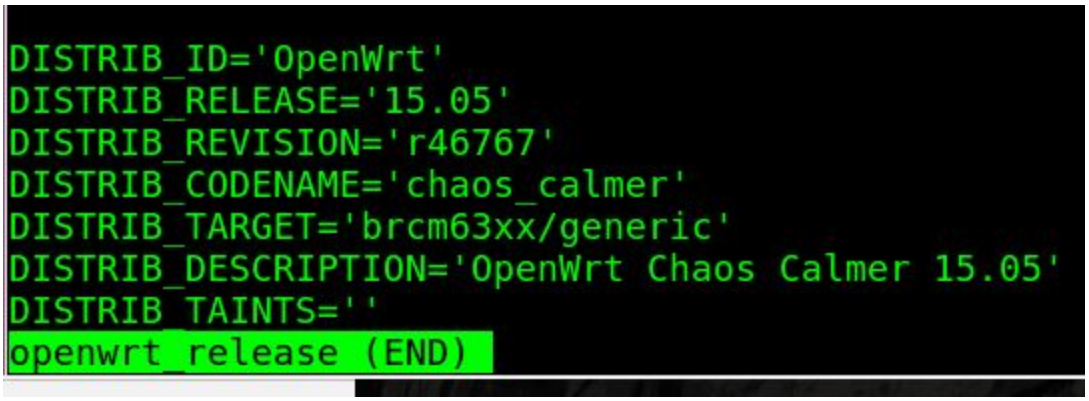
Also failsafe mode information could be found in this file @ /etc/banner.failsafe

4. Try and find the following items:

a. What is most likely the device vendor's name for the assigned firmware?

As binwalk gave the hint of broadcom, I googled broadcom ubiquiti wireless router and the first link that pooped up was this . So, I believe that the most likely device vendor's name is 'UBIQUITI' and I also got the suggested vendor name in /usr/share/libiwinfo folder, as shown in below screenshot.



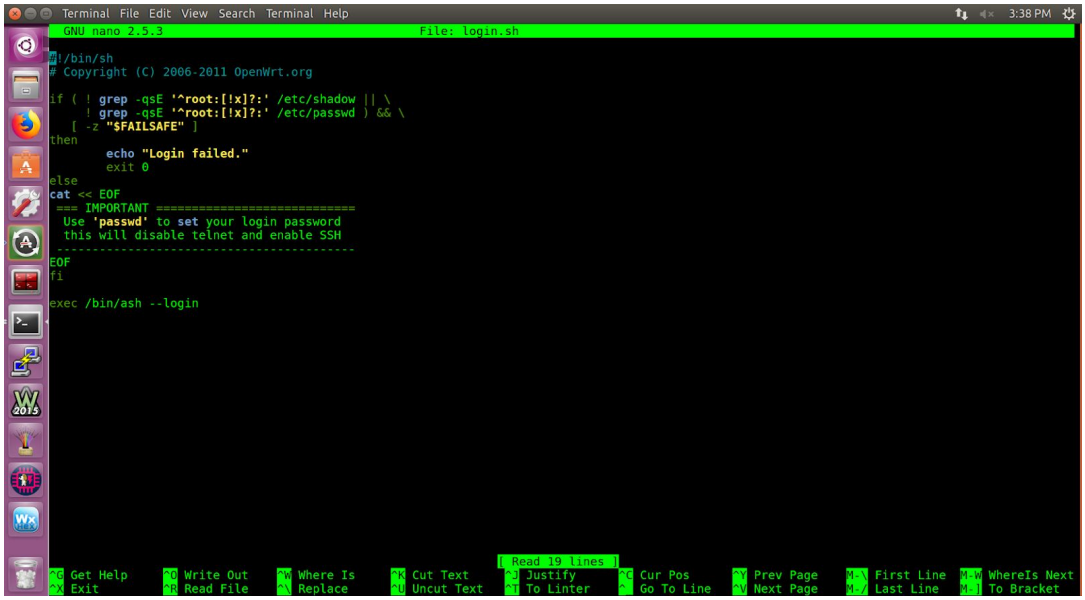


b. What is most likely the device name for which this firmware is meant to be run on? **WiFi router**

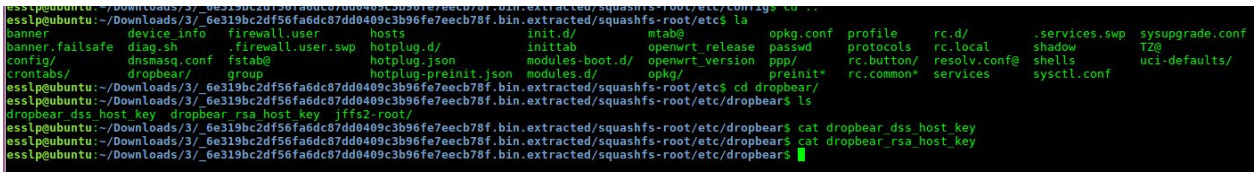
c. Hard coded passwords

passwd

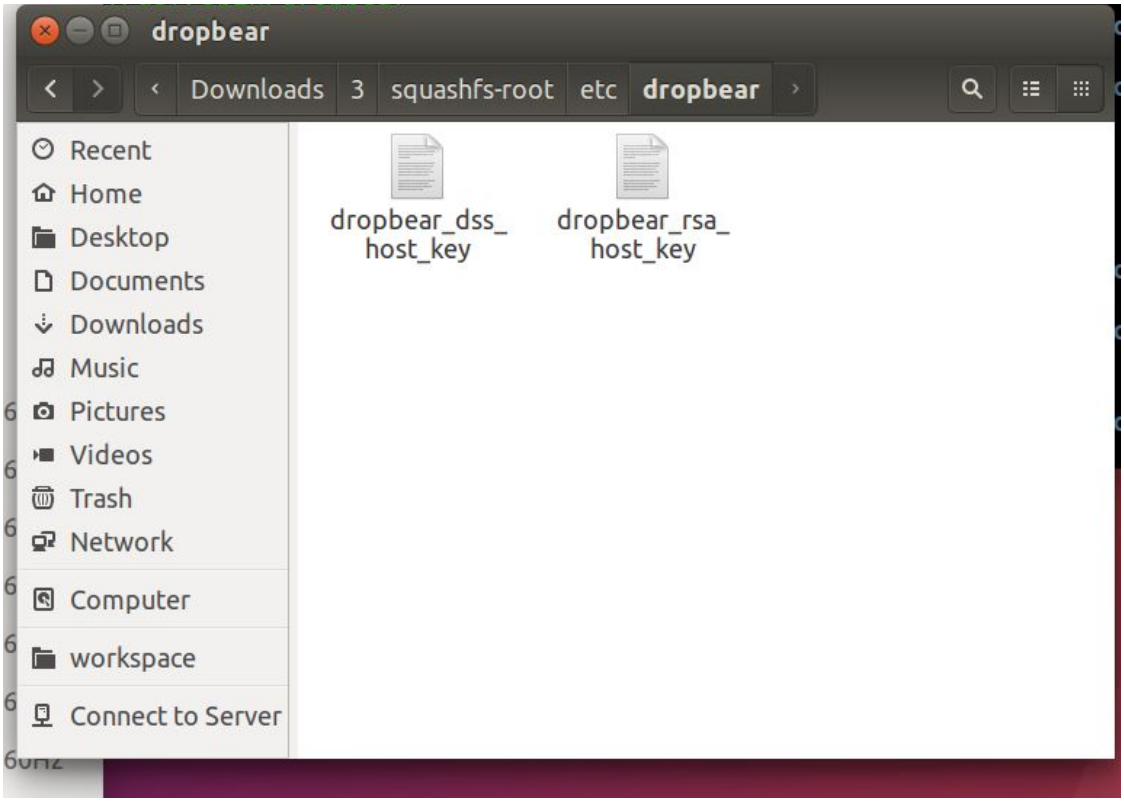
found the file name and file path of the file containing hardcoded password by using the command**: grep -iRn 'password'**



a. Certificates for signing/encryption



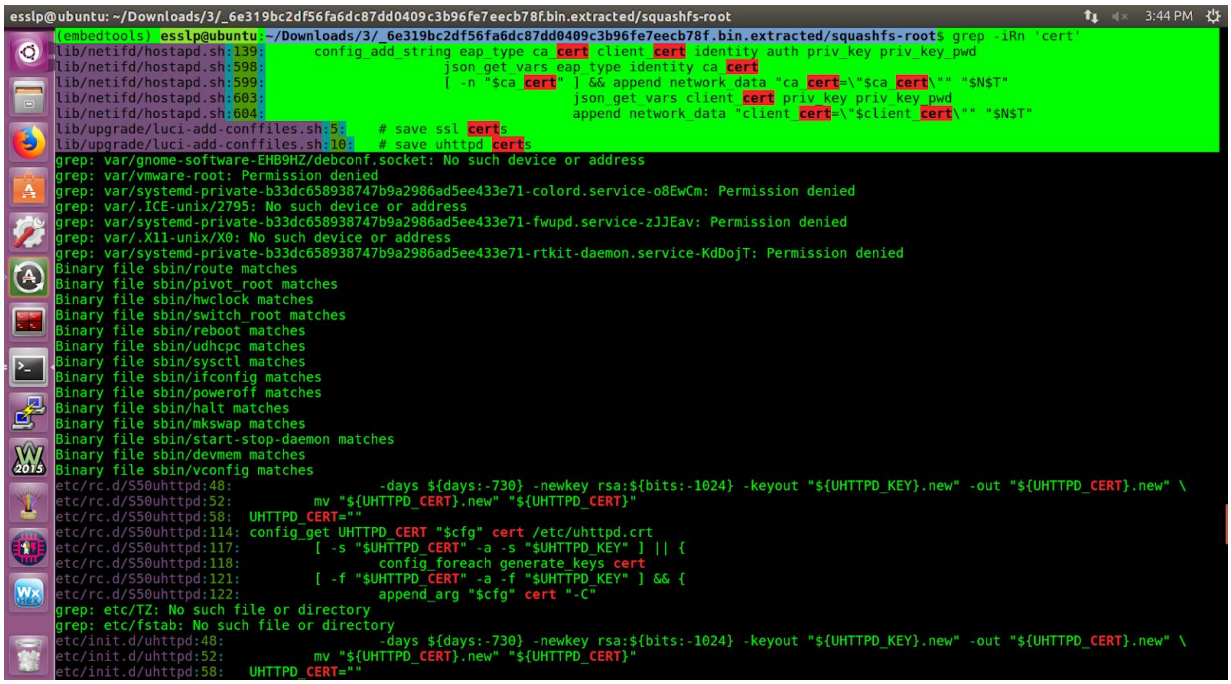
The two files inside dropbear which shall content the certificate for signing/encryption information were empty.



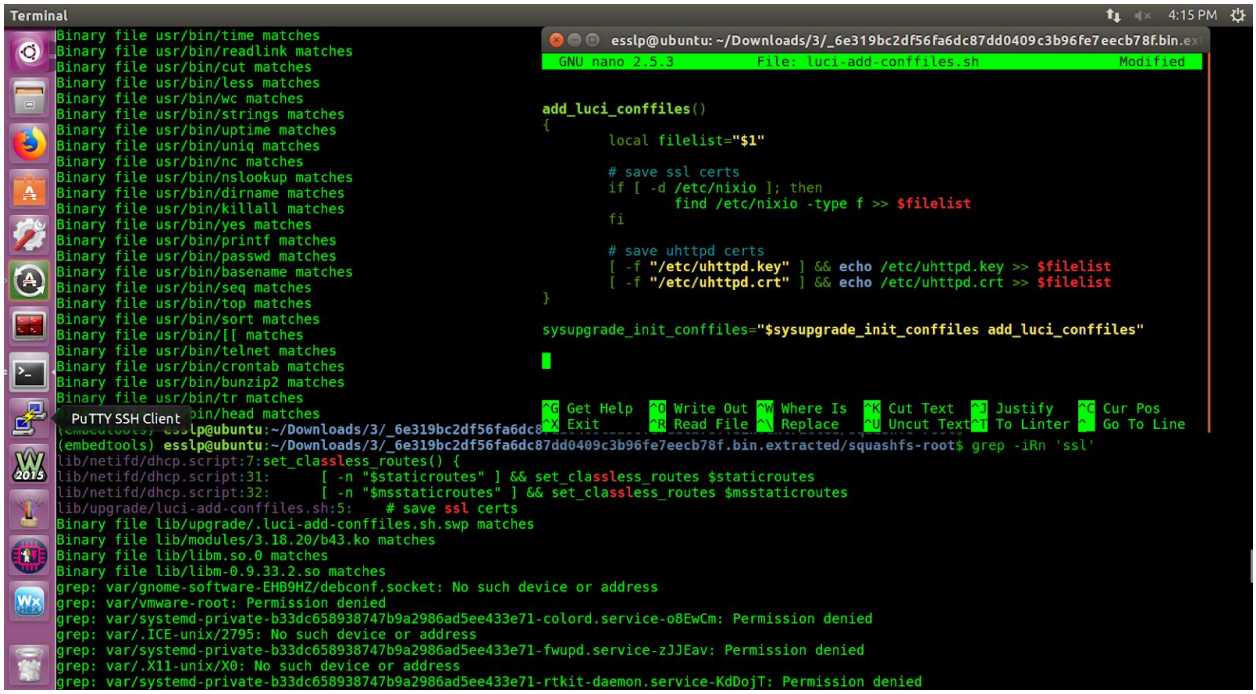
/etc/config/uhttpd

```
# Certificate defaults for px5g key generator
config cert px5g

        # Validity time
        option days             730

        # RSA key size
        option bits             1024

        # Location
        option country          ZZ
        option state            Somewhere
        option location         Uknown

        # Common name
        option commonname       OpenWrt
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/config$
```

Above screenshot shows the validity time for the certificate, RSA key size, location details.



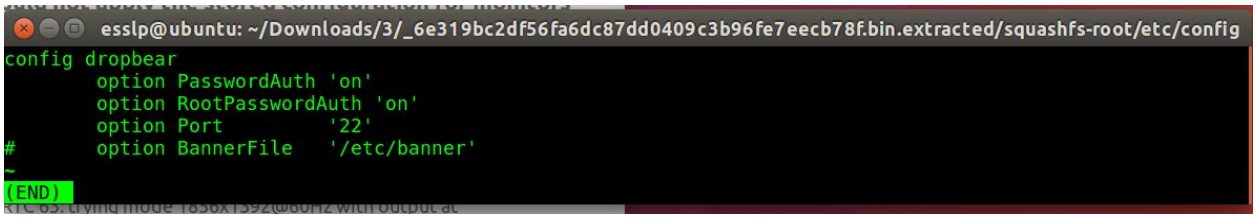
This displays the path for save ssl cert,



b. Configuration files for secure utilities such as OpenSSL, OpenSSH, tunneling protocols such as OpenVPN etc.

Openwrt uses Dropbear for SSH (hence, it can be considered as one of the secured utilities.) More information about dropbear, I gained from this source

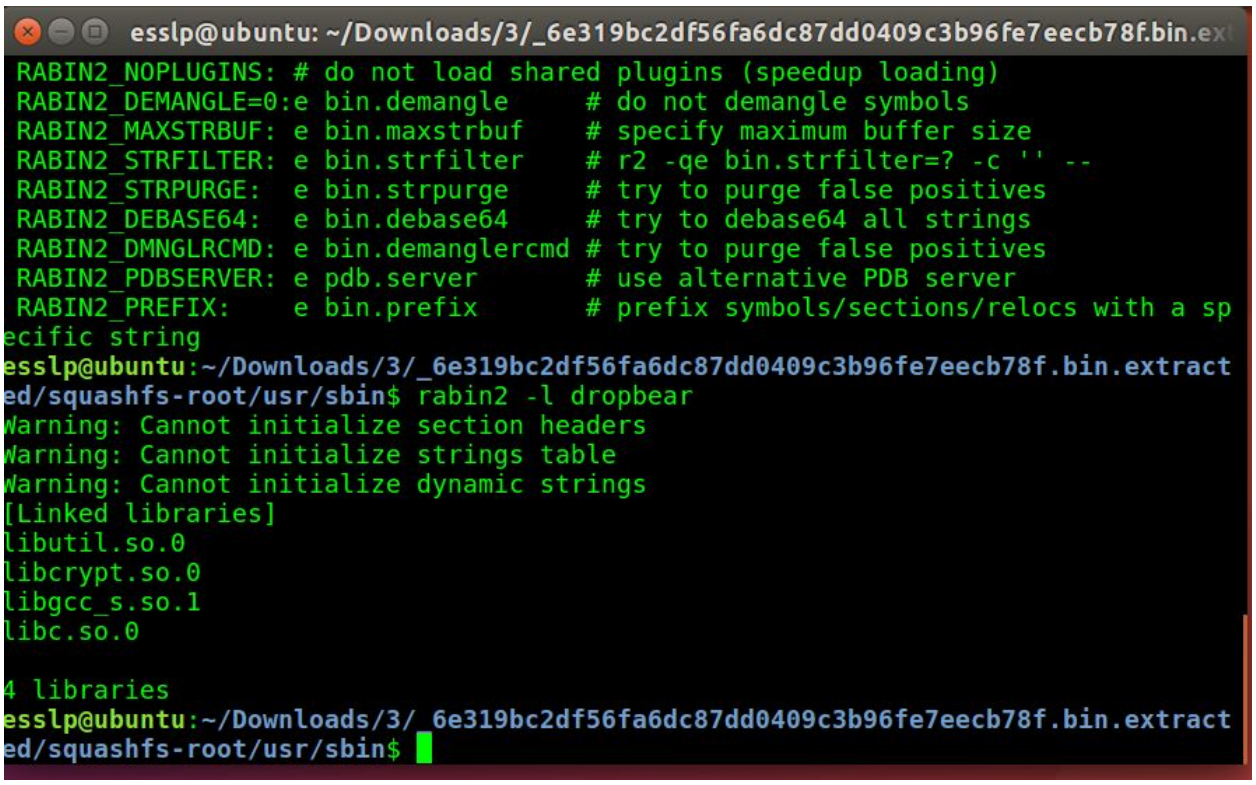
/etc/config/dropbear



This is the content of Dropbear file: It shows that password authentication is turned ON, Root password Authentication is also ON, The port for SSH is 22 and banner file is stored in location '/etc/banner'

c. Libraries associated with the secure utilities found in the step above.

Below are the 4 linked libraries attached to dropbear found using rabin2 utility.



d. Version numbers for the libraries you found in the step above?

Libutil version number : 0.9.33.2.so

```
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ grep -iRn libutil
Binary file lib/libutil-0.9.33.2.so matches
Binary file lib/libutil.so.0 matches
grep: var/vmware-root: Permission denied
grep: var/.ICE-unix/3141: No such device or address
grep: var/systemd-private-e0b6a3dabcf54f6baabab614df150138-rtkit-daemon.service-eOKIKb: Permission denied
grep: var/gnome-software-1H2WIZ/debconf.socket: No such device or address
grep: var/.X11-unix/X0: No such device or address
grep: var/systemd-private-e0b6a3dabcf54f6baabab614df150138-colord.service-Aw4gxA: Permission denied
grep: var/systemd-private-e0b6a3dabcf54f6baabab614df150138-fwupd.service-d6NPAk: Permission denied
grep: etc/TZ: No such file or directory
grep: etc/fstab: No such file or directory
grep: etc/resolv.conf: No such file or directory
grep: etc/ppp/resolv.conf: No such file or directory
usr/lib/opkg/info/libc.list:3:/lib/libutil-0.9.33.2.so
usr/lib/opkg/info/libc.list:6:/lib/libutil.so.0
Binary file usr/sbin/pppd matches
Binary file usr/sbin/dropbear matches
Binary file usr/bin/scp matches
Binary file usr/bin/dbclient matches
Binary file usr/bin/ssh matches
Binary file usr/bin/dropbearkey matches
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$
```

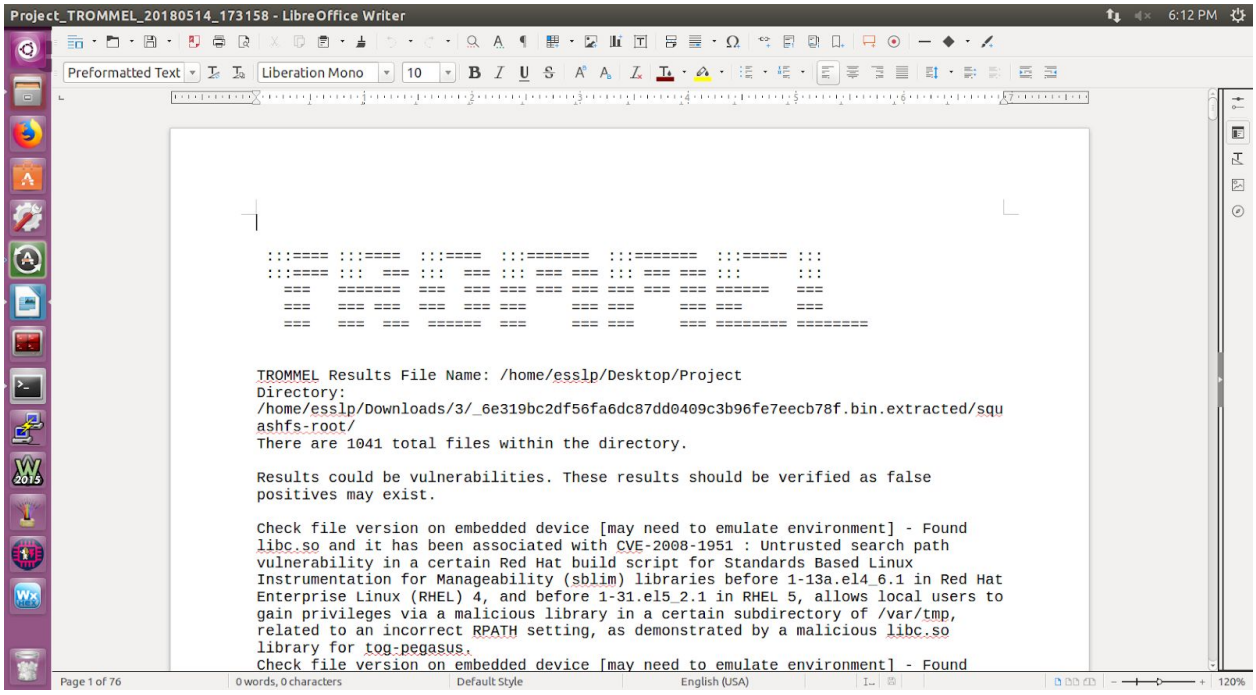
Libcrypt version using command **grep -iRn libcrypt**

```
usr/lib/opkg/info/libc.list:1:/lib/libcrypt-0.9.33.2.so
usr/lib/opkg/info/libc.list:11:/lib/libcrypt.so.0
```

5. What are some of the known vulnerabilities in the utilities you found in Step 5.f and 5.g? You can use existing tools such as trommel or look up the specific versions of those utilities on the internet at sites such as:

```
esslp@ubuntu: ~/workspace/embedtools/trommel
esslp@ubuntu:~$ workon embedtools
(embedtools) esslp@ubuntu:~/workspace/embedtools$ cd trommel/
(embedtools) esslp@ubuntu:~/workspace/embedtools/trommel$ ~/workspace/embedtools/trommel/trommel.py -p /home/esslp/Downloads/3
3/     3.zip
(embedtools) esslp@ubuntu:~/workspace/embedtools/trommel$ ~/workspace/embedtools/trommel/trommel.py -p /home/esslp/Downloads/3/_6e319bc2df56fa
6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/ -o /home/esslp/Desktop/Project
DB:  /home/esslp/workspace/embedtools/trommel/vfeed.db

TROMMEL is working to sift through the directory of files.
Results will be saved to '/home/esslp/Desktop/Project_TROMMEL_20180514_173158'

(embedtools) esslp@ubuntu:~/workspace/embedtools/trommel$
```

This are some of the vulnerabilities found using Trommel.



These are the two vulnerabilities found online with CVE search on Dropbear.

CVE-2017-9078 : This vulnerability allows post authentication root code execution because of a double free in cleanup of TCP listeners when the -a option is enabled. Meaning, it could result in denial of service by an authenticated user if Dropbear is running with the -a option.

CVE-2017-9079 : This vulnerability allows local users to read certain files as root, if the file has the authorized_keys file format with a command = option. More precisely, a local information leak is leveraged in parsing the authorized_keys file.

These two vulnerabilities were also found in trommel. As it is 2017, I believe that it is pretty new vulnerability.

```
s-root/usr/lib/opkg/info/ubusd.prerm, Keyword Hits in File: 1
Check file version on embedded device [may need to emulate environment] - Found
dropbear and it has been associated with CVE-2017-9079 : Dropbear before 2017.75 might
allow local users to read certain files as root, if the file has the authorized_keys
file format with a command= option. This occurs because ~/.ssh/authorized_keys is read
with root privileges and symlinks are followed.
Check file version on embedded device [may need to emulate environment] - Found
dropbear and it has been associated with CVE-2017-9078 : The server in Dropbear before
2017.75 might allow post-authentication root remote code execution because of a double
free in cleanup of TCP listeners when the -a option is enabled.
```

6. What would your next steps be as an attacker knowing what you know from the steps above?

For CVE-2017-9078 : An attacker can gain Admin level privileges leveraging this vulnerability. Once, an attacker gain Admin rights he/she could pretty much owns the firmware. Could modify passwords, could delete/modify config files. Create a backdoor or add him/herself as an user.

For CVE-2017-9079 : An local user could read files of higher privileges, This gives local user chance to leak information, It could also lead to more hazardous when an attacker gains access to local user account and leveraging this vulnerability could read confidential information about firmware, config files, encryption keys, etc.

7 What steps would you take to prevent the vulnerabilities you found based on what you know from steps above?

For CVE-2017-9078 : Generate hostkeys with dropbearkey atomically and flush to disk with fsync. Upgrade version.

Reference:

http://lists.ucc.gu.uwa.edu.au/pipermail/dropbear/2017q2/001985.html
https://www.debian.org/security/2017/dsa-3859
https://security-tracker.debian.org/tracker/CVE-2017-9079
https://www.cvedetails.com/cve/CVE-2017-9078/
https://github.com/openwrt/packages/issues/2562
https://openwrt.org/releases/15.05/notes-15.05.1
https://lwn.net/Articles/649870/
https://security.stackexchange.com/questions/12131/dropbear-ssh-server-use-after-free-vulnerability
https://secure.ucc.asn.au/hg/dropbear/rev/818108bf7749