

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>**

```

Terminal
esslp@ubuntu: ~/Downloads
esslp@ubuntu:~/Downloads$ cd Downloads/
esslp@ubuntu:~/Downloads$ ls
3.zip
dbeaver-ce_latest_amd64.deb  gede-2.4.3/      section7 (2)/
embed@192.168.7.2/          gede-2.4.3.tar.xz section7.zip
esslp@ubuntu:~/Downloads$ unzip 3.zip
Archive: 3.zip
  creating: 3/
    inflating: 3/6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin
esslp@ubuntu:~/Downloads$ ls
3/
3.zip
dbeaver-ce_latest_amd64.deb  gede-2.4.3/      section7 (2)/
embed@192.168.7.2/          gede-2.4.3.tar.xz section7.zip
esslp@ubuntu:~/Downloads$ binwalk 3
3/      3.zip
esslp@ubuntu:~/Downloads$ binwalk 3/
esslp@ubuntu:~/Downloads$ binwalk 3/6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin
DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0             0x0          Broadcom 96345 firmware header, header size: 256, firmware v
ersion: "8", board id: "6368M-1541N", -CRC32 header checksum: 0x521BA623, -CRC32 data chec
ksum: 0xF5919A10
1174938      0x11ED9A     Squashfs filesystem, little endian, version 4.0, compression
:xz, size: 2318674 bytes, 1131 inodes, blocksize: 262144 bytes, created: 2015-09-11 16:31:
53
esslp@ubuntu:~/Downloads$

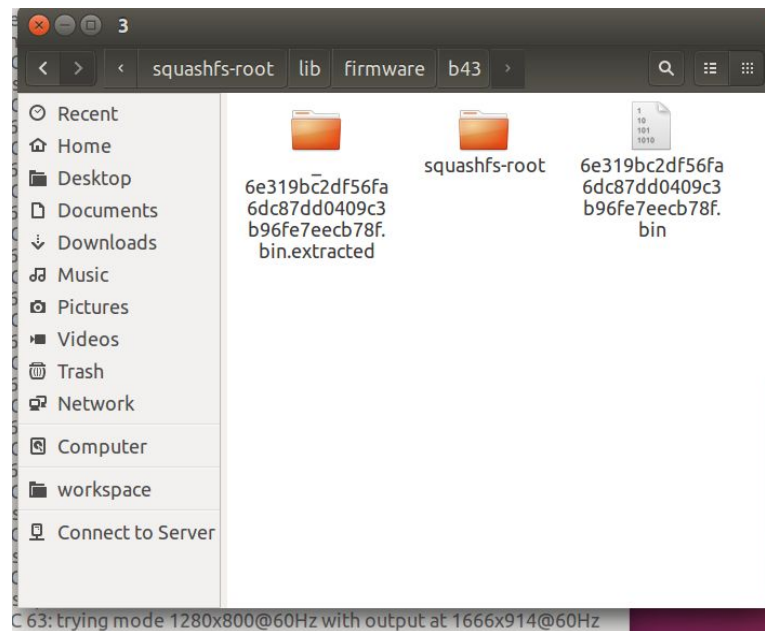
```

```

(embedtools) esslp@ubuntu:~/Downloads$ binwalk -Me 3.zip
Scan Time:      2018-05-13 09:22:00
Target File:    /home/esslp/Downloads/3.zip
MD5 Checksum:   b9cae673ce4775bf6e89969a6184b668
Signatures:     386
DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0             0x0          Zip archive data, at least v1.0 to extract, name: 3/
32            0x28         Zip archive data, at least v2.0 to extract, compressed size: 3508313, uncompressed size: 16777472, name: 3/6e319bc
2df56fa6dc87dd0409c3b96fe7eeb78f.bin
3508633      0x358999     End of Zip archive, footer length: 22
Scan Time:      2018-05-13 09:22:03
Target File:    /home/esslp/Downloads/_3.zip.extracted/3/6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin
MD5 Checksum:   675d1536817818e0366fe2e35c1b1525
Signatures:     386
DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0             0x0          Broadcom 96345 firmware header, header size: 256, firmware version: "8", board id: "6368M-1541N", -CRC32 header ch
ecksum: 0x521BA623, -CRC32 data checksum: 0xF5919A10
1174938      0x11ED9A     Squashfs filesystem, little endian, version 4.0, compression:xz, size: 2318674 bytes, 1131 inodes, blocksize: 2621
44 bytes, created: 2015-09-11 16:31:53
Trash
(embedtools) esslp@ubuntu:~/Downloads$

```

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 date



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

```

esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ ls
3/
3.zip
dbeaver-ce_latest_amd64.deb
gede-2.4.3/
gede-2.4.3.tar.xz
section7/
section7 (2)/
section7.zip
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ ls
11ED9A.squashfs
squashfs-root/
uci.sh
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ cd squashfs-root/
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ ls
bin/
etc/
lib/
overlay/
rom/
sbin/
tmp/
var/
dev/
init*
mnt/
proc/
root/
sys/
usr/
www/
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$

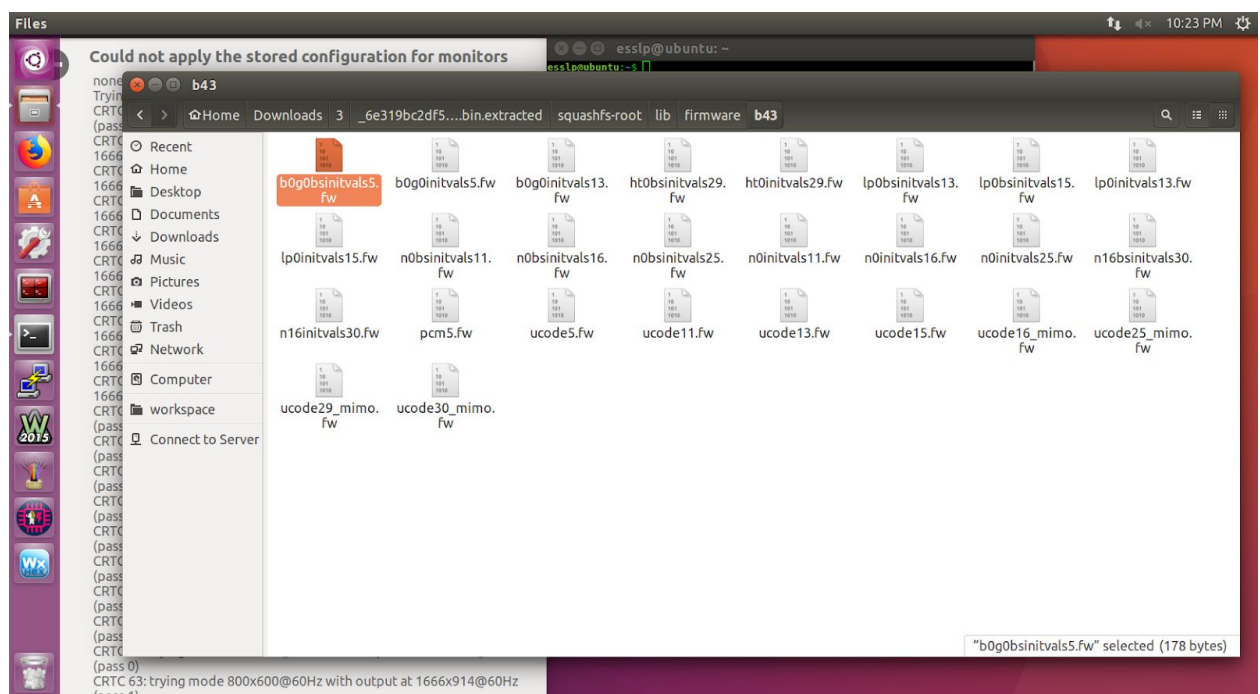
```

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

```

ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/bin$ cd ..
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root$ cd lib/
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/lib$ cd firmware/
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/lib/firmware$ ls
b43/
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/lib/firmware$ cd b43/
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/lib/firmware/b43$ ls
b0g0bsinitvals5.fw  ht0bsinitvals29.fw  lp0bsinitvals13.fw  n0bsinitvals11.fw  n16bsinitvals30.fw  ucode11.fw  ucode16_mimo.fw  ucode30_mimo.fw
lp0bsinitvals15.fw  ht0bsinitvals29.fw  lp0bsinitvals13.fw  n0bsinitvals16.fw  n0bsinitvals16.fw  n16initvals30.fw  ucode13.fw  ucode25_mimo.fw  ucode5.fw
n0bsinitvals11.fw  lp0bsinitvals15.fw  lp0bsinitvals13.fw  n0bsinitvals15.fw  n0bsinitvals25.fw  pcm5.fw  ucode15.fw  ucode29_mimo.fw
ssslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/lib/firmware/b43$

```



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

```

(embedtools) esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/bin$ file ./busybox
./busybox: ELF 32-bit MSB executable, MIPS, MIPS32 version 1, dynamically linked, interpreter /lib/ld-ucLibc.so.0, corrupted section header si
(embedtools) esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeb78f.bin.extracted/squashfs-root/bin$

```

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

```

esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/lib
# /bin/sh
# Copyright (C) 2007 OpenWrt.org
#
board_id=""
sys_mtd_part=""
ifname=""

bcm63xx_dt_detect() {
    local board_name

    case "$1" in
        "ADB P.DG A4001N")
            board_name="a4001n"
            ;;
        "ADB P.DG A4001N1")
            board_name="a4001n1"
            ;;
        "Alcatel RG100A")
            board_name="rg100a"
            ;;
        "ASMAX AR 1004g")
            board_name="ar100g"
            ;;
        "Belkin F5D7633")
            board_name="f5d7633"
            ;;
        "Broadcom 96348R reference board")
            board_name="bcm96348r"
            ;;
        "Broadcom BCM96318REF reference board")
            board_name="bcm96318ref"
            ;;
        "Broadcom BCM96318REF_P300 reference board")
            board_name="bcm96318ref_p300"
            ;;
        "Broadcom BCM963268BU_P300 reference board")
            board_name="bcm963268bu_p300"
            ;;
    esac
}

```

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

```

===== FAILSAFE MODE active =====
special commands:
* firstboot          reset settings to factory defaults
* mount_root         mount root-partition with config files

after mount_root:
* passwd             change root's password
* /etc/config         directory with config files

for more help see:
http://wiki.openwrt.org/doc/howto/generic.failsafe
=====

banner.failsafe (END)

```

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 popped 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.


```

Terminal File Edit View Search Terminal Help
ess@ubuntu:~/Downloads/3_6e319c2df56fadc37d0409c3b96fe7ecb78f.bin.extracted/squashfs-root/usr/share/libinfo$ grep -r -o 'vendor name'
hardware.txt: # expower offset | frequency offset | "vendor name" | "device name"
ess@ubuntu:~/Downloads/3_6e319c2df56fadc37d0409c3b96fe7ecb78f.bin.extracted/squashfs-root/usr/share/libinfo$ cat hardware.txt
# libinfo hardware database
# vendor id | device id | subsystem vendor id | subsystem device id |
# expower offset | frequency offset | "vendor name" | "device name"
0xffff 0xffff 0xffff 0xb202 0 0 "Ubiquiti" "PowerStation2 (18V)"
0xffff 0xffff 0xffff 0xb302 0 0 "Ubiquiti" "PowerStation2 (180)"
0xffff 0xffff 0xffff 0xb305 0 0 "Ubiquiti" "PowerStation5 (220)"
0xffff 0xffff 0xffff 0xb305 0 0 "Ubiquiti" "PowerStation5 (EXT)"
0xffff 0xffff 0xffff 0xc302 0 0 "Ubiquiti" "PicoStation2"
0xffff 0xffff 0xffff 0xc302 10 0 "Ubiquiti" "PicoStation2 ap"
0xffff 0xffff 0xffff 0xa105 0 0 "Ubiquiti" "WispStation5"
0xffff 0xffff 0xffff 0xa002 10 0 "Ubiquiti" "NanoStation2"
0xffff 0xffff 0xffff 0xc005 5 0 "Ubiquiti" "LiteStation5"
0xffff 0xffff 0xffff 0xc002 10 0 "Ubiquiti" "NanoStation2"
0xffff 0xffff 0xffff 0xc005 5 0 "Ubiquiti" "NanoStation2"
0xffff 0xffff 0xffff 0xc102 10 0 "Ubiquiti" "NanoStation Loco2"
0xffff 0xffff 0xffff 0xc105 5 0 "Ubiquiti" "NanoStation Loco5"
0xffff 0xffff 0xffff 0xc202 10 0 "Ubiquiti" "Bullet2"
0xffff 0xffff 0xffff 0xc205 5 0 "Ubiquiti" "Bullet5"
0x168c 0x001b 0x0777 0x3002 10 0 "Ubiquiti" "XR2"
0x168c 0x001b 0x0777 0x3002 10 0 "Ubiquiti" "XR2"
0x168c 0x001b 0x0777 0x3002 10 0 "Ubiquiti" "XR2.3"
0x168c 0x001b 0x0777 0x3002 10 0 "Ubiquiti" "XR2.6"
0x168c 0x001b 0x0777 0x3002 10 0 "Ubiquiti" "XR3-2.6"
0x168c 0x001b 0x0777 0x3003 10 0 "Ubiquiti" "XR3-3.6"
0x168c 0x001b 0x0777 0x3003 10 0 "Ubiquiti" "XR3"
0x168c 0x001b 0x0777 0x3004 10 0 "Ubiquiti" "XR4"
0x168c 0x001b 0x0777 0x3005 10 0 "Ubiquiti" "XR5"
0x168c 0x001b 0x0777 0x3005 10 0 "Ubiquiti" "XR5"
0x168c 0x001b 0x0777 0x3007 10 0 "Ubiquiti" "XR7"
0x168c 0x001b 0x0777 0x3009 10 -1520 "Ubiquiti" "XR9"
0x168c 0x001b 0x168c 0x2063 0 0 "Atheros" "AR5413"
0x168c 0x0013 0x168c 0x2042 1 0 "Ubiquiti" "SR1"
0x168c 0x0013 0x0777 0x2041 10 0 "Ubiquiti" "SR2"
0x168c 0x0013 0x0777 0x2004 6 0 "Ubiquiti" "SR4"
0x168c 0x0013 0x0777 0x2004 6 0 "Ubiquiti" "SR4"
0x168c 0x0013 0x0777 0x1004 6 0 "Ubiquiti" "SR4C"
0x168c 0x0013 0x0777 0x1004 6 0 "Ubiquiti" "SR4C"
0x168c 0x0013 0x168c 0x2042 7 0 "Ubiquiti" "SR5"
0x168c 0x0013 0x0777 0x2009 12 -1500 "Ubiquiti" "SR9"
0x168c 0x0027 0x168c 0x2002 7 0 "Ubiquiti" "SR71A"
0x168c 0x0027 0x0777 0x0002 7 0 "Ubiquiti" "SR71"
0x168c 0x0029 0x0777 0x4005 7 0 "Ubiquiti" "SR71-15"
0x168c 0x0029 0x0777 0xe302 12 0 "Ubiquiti" "PicoStation MS" /* ToDo: confirm offset */
0x168c 0x0029 0x0777 0xe302 12 0 "Ubiquiti" "NanoStation MS" /* ToDo: confirm offset */
0x168c 0x0029 0x0777 0xe005 5 0 "Ubiquiti" "NanoStation MS" /* ToDo: confirm offset */
0x168c 0x0029 0x0777 0xe202 12 0 "Ubiquiti" "Bullet MS"
0x168c 0x0029 0x0777 0xe005 5 0 "Ubiquiti" "Bullet MS"
0x168c 0x0029 0x0777 0xe345 0 0 "Ubiquiti" "WispStation MS" /* ToDo: confirm offset */
0x168c 0x0029 0x168c 0xa004 0 0 "Atheros" "AR0200"
0x168c 0x0029 0x168c 0xa005 0 0 "Atheros" "AR0201"
0x168c 0x0029 0x168c 0xa003 0 0 "Atheros" "AR0200"
0x168c 0x0029 0x168c 0xa001 0 0 "Atheros" "AR0205"
0x168c 0x0033 0x168c 0xa120 0 0 "Atheros" "AR0580"
0x168c 0x0033 0x168c 0xa136 0 0 "Atheros" "AR0580"
0x168c 0x003c 0x0000 0x0000 0 0 "Qualcomm Atheros" "QCA9880"
0x1814 0x3050 0x1814 0x0005 0 0 "Realink" "Rt3050"
0x1814 0x3052 0x1814 0x0008 0 0 "Realink" "Rt3052"
0x1814 0x3052 0x1814 0x000c 0 0 "Realink" "Rt3052"
0x11ab 0x2255 0x11ab 0x0000 0 0 "Marvell" "88W8064"

```

```

DISTRIB_ID='OpenWrt'
DISTRIB_RELEASE='15.05'
DISTRIB_REVISION='r46767'
DISTRIB_CODENAME='chaos_calmer'
DISTRIB_TARGET='bcm63xx/generic'
DISTRIB_DESCRIPTION='OpenWrt Chaos Calmer 15.05'
DISTRIB_TAINTS=''
openwrt release (END)

```

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'**

```

GNU nano 2.5.3 File: login.sh
#!/bin/sh
# Copyright (C) 2006-2011 OpenWrt.org

if ( ! grep -qsE '^root:[!x]?:' /etc/shadow || \
    ! grep -qsE '^root:[!x]?:' /etc/passwd ) && \
then
    echo "Login failed."
    exit 0
else
    cat << EOF
    == IMPORTANT ==
    Use 'passwd' to set your login password
    this will disable telnet and enable SSH
    EOF
fi

exec /bin/ash --login
  
```

```

esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/bin
Use "fg" to return to nano.

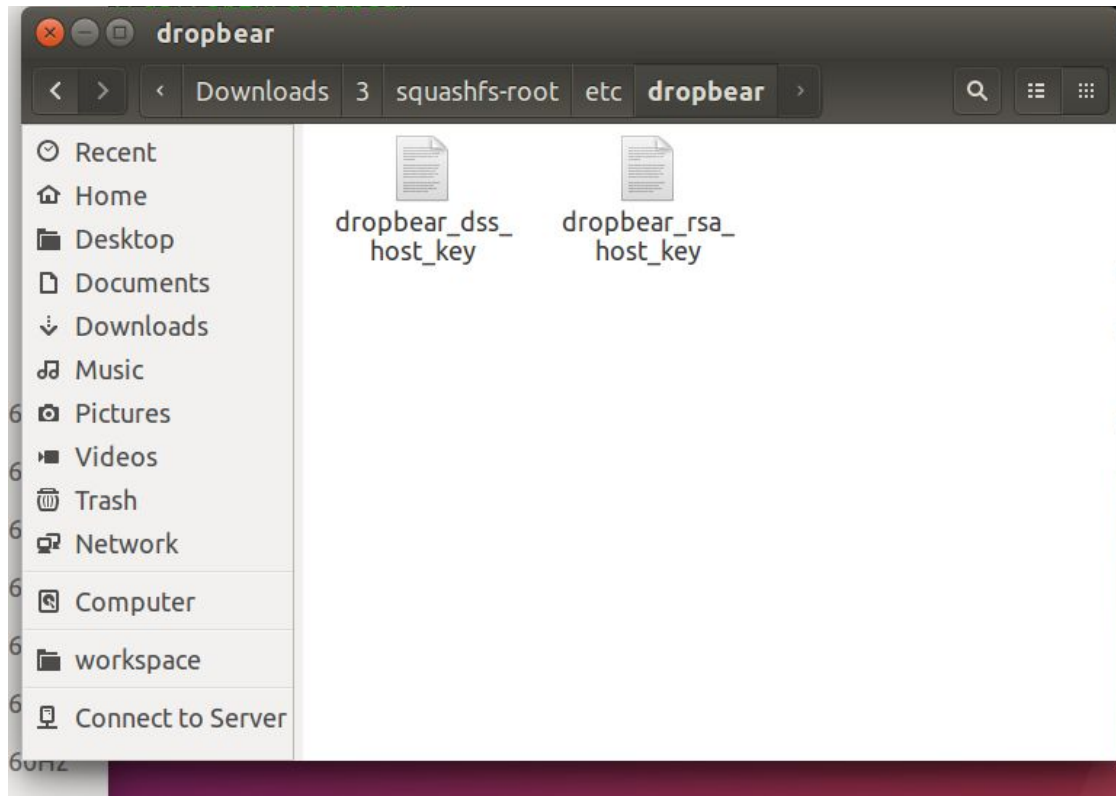
[10]- Stopped nano login.sh
(embedtools) esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/bin$ grep -iRn 'password'
Binary file touch matches
Binary file ping matches
Binary file fgrep matches
Binary file egrep matches
Binary file netmsg matches
Binary file fsync matches
Binary file chmod matches
Binary file zcat matches
Binary file true matches
Binary file gunzip matches
Binary file echo matches
Binary file df matches
Binary file mktmp matches
Binary file cp matches
Binary file mkdir matches
Binary file mount matches
Binary file chgrp matches
Binary file ping6 matches
Binary file uname matches
Binary file grep matches
Binary file ls matches
Binary file busybox matches
Binary file lock matches
Binary file rmdir matches
Binary file date matches
Binary file ps matches
Binary file false matches
Binary file sync matches
Binary file pidof matches
Binary file sed matches
Binary file mknd matches
Binary file ln matches
Binary file rm matches
Binary file chown matches
Binary file nice matches
login.sh:13: Use 'passwd' to set your login password
Trash: sh matches
Binary file vi matches
  
```

a. Certificates for signing/encryption

```

esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc$ cd ..
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc$ ls
banner device info firewall.user hosts init.d mtab8 profile rc.d .services.swp sysupgrade.conf
banner.failsafe diag.sh .firewall.user.swp hotplug.d/ inittab openwrt release passwd protocols rc.local shadow TZ@
config/ dnsmasq.conf fstab@ hotplug.json modules-boot.d/ openwrt_version ppp/ rc.button/ resolv.conf@ shells uci-defaults/
crontabs/ dropbear/ group hotplug-preinit.json modules.d/ opkg/ preinit* rc.common* services sysctl.conf
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/dropbear$ ls
dropbear_dss_host_key dropbear_rsa_host_key jffs2-root/
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/dropbear$ cat dropbear_dss_host_key
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/dropbear$ cat dropbear_rsa_host_key
esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/dropbear$
  
```

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      Unknown

    # Common name
    option commonname    OpenWrt
esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eeeb78f.bin.extracted/squashfs-root/etc/config$
```

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

```

esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root
[embedtools] esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ grep -iRn 'cert'
lib/netifd/hostapd.sh:139:      config add string eap_type ca $cert client $cert identity auth priv_key priv_key_pwd
lib/netifd/hostapd.sh:598:      json get vars eap_type identity ca $cert
lib/netifd/hostapd.sh:599:      [ -n "$ca $cert" ] && append network_data "ca $cert \"$sca $cert\"" "$N$T"
lib/netifd/hostapd.sh:603:      json get vars client $cert priv_key priv_key_pwd
lib/netifd/hostapd.sh:604:      append network_data "client $cert=\"$sclient $cert\"" "$N$T"
lib/upgrade/luci-add-conffiles.sh:5: # save ssl certs
lib/upgrade/luci-add-conffiles.sh:10: # save uhttpd certs
grep: var/gnome-software-EHB9HZ/debconf.socket: No such device or address
grep: var/vmware-root: Permission denied
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-colord.service-o8EwCm: Permission denied
grep: var/.ICE-unix/2795: No such device or address
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-fwupd.service-zJJEv: Permission denied
grep: var/.X11-unix/X0: No such device or address
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-rtkit-daemon.service-KdDojT: Permission denied
Binary file sbin/route matches
Binary file sbin/pivot root matches
Binary file sbin/hwclock matches
Binary file sbin/switch root matches
Binary file sbin/reboot matches
Binary file sbin/udhcpc matches
Binary file sbin/sysctl matches
Binary file sbin/ifconfig matches
Binary file sbin/poweroff matches
Binary file sbin/halt matches
Binary file sbin/mkswap matches
Binary file sbin/start-stop-daemon matches
Binary file sbin/devmem matches
Binary file sbin/vconfig matches
etc/rc.d/s50uhttpd:48:      -days $(days:-730) -newkey rsa:$(bits:-1024) -keyout "${UHTTPD_KEY}.new" -out "${UHTTPD_CERT}.new" \
etc/rc.d/s50uhttpd:52:      mv "${UHTTPD_CERT}.new" "${UHTTPD_CERT}"
etc/rc.d/s50uhttpd:58:      UHTTPD_CERT=""
etc/rc.d/s50uhttpd:114:      config_get UHTTPD_CERT "$cfg" cert /etc/uhttpd.crt
etc/rc.d/s50uhttpd:117:      [ -s "$UHTTPD_CERT" -a -s "$UHTTPD_KEY" ] || {
etc/rc.d/s50uhttpd:118:          config foreach generate_keys cert
etc/rc.d/s50uhttpd:121:      [ -f "$UHTTPD_CERT" -a -f "$UHTTPD_KEY" ] && {
etc/rc.d/s50uhttpd:122:          append_arg "$cfg" cert "-C"
grep: etc/TZ: No such file or directory
grep: etc/fstab: No such file or directory
etc/init.d/uhttpd:48:      -days $(days:-730) -newkey rsa:$(bits:-1024) -keyout "${UHTTPD_KEY}.new" -out "${UHTTPD_CERT}.new" \
etc/init.d/uhttpd:52:      mv "${UHTTPD_CERT}.new" "${UHTTPD_CERT}"
etc/init.d/uhttpd:58:      UHTTPD_CERT=""

```

This displays the path for save ssl cert,

```

Terminal
Binary file usr/bin/time matches
Binary file usr/bin/readlink matches
Binary file usr/bin/cut matches
Binary file usr/bin/wc matches
Binary file usr/bin/strings matches
Binary file usr/bin/uptime matches
Binary file usr/bin/uniq matches
Binary file usr/bin/nc matches
Binary file usr/bin/nslookup matches
Binary file usr/bin/dirname matches
Binary file usr/bin/killall matches
Binary file usr/bin/yes matches
Binary file usr/bin/printenv matches
Binary file usr/bin/passwd matches
Binary file usr/bin/basename matches
Binary file usr/bin/seq matches
Binary file usr/bin/top matches
Binary file usr/bin/sort matches
Binary file usr/bin/[[ matches
Binary file usr/bin/telnet matches
Binary file usr/bin/crontab matches
Binary file usr/bin/bunzip2 matches
Binary file usr/bin/tr matches
Binary file bin/head matches
[embedtools] esslp@ubuntu:~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root$ grep -iRn 'ssl'
lib/netifd/dhcp.script:7:set_classless_routes() {
lib/netifd/dhcp.script:31:    [ -n "$staticroutes" ] && set_classless_routes $staticroutes
lib/netifd/dhcp.script:32:    [ -n "$msstaticroutes" ] && set_classless_routes $msstaticroutes
lib/upgrade/luci-add-conffiles.sh:5: # save ssl certs
Binary file lib/upgrade/luci-add-conffiles.sh.swp matches
Binary file lib/modules/3.18.20/b43.ko matches
Binary file lib/libm.so.0 matches
Binary file lib/libm-0.9.33.2.so matches
grep: var/gnome-software-EHB9HZ/debconf.socket: No such device or address
grep: var/vmware-root: Permission denied
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-colord.service-o8EwCm: Permission denied
grep: var/.ICE-unix/2795: No such device or address
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-fwupd.service-zJJEv: Permission denied
grep: var/.X11-unix/X0: No such device or address
grep: var/systemd-private-b33dc658938747b9a2986ad5ee433e71-rtkit-daemon.service-KdDojT: Permission denied

```

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

```

esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/etc/config
config dropbear
    option PasswordAuth 'on'
    option RootPasswordAuth 'on'
    option Port '22'
#    option BannerFile '/etc/banner'
~
(END)

```

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.

```

esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/usr/sbin$ rabin2 -l dropbear
RABIN2_NOPLUGINS: # do not load shared plugins (speedup loading)
RABIN2_DEMANGLE=0: e bin.demangle # do not demangle symbols
RABIN2_MAXSTRBUF: e bin.maxstrbuf # specify maximum buffer size
RABIN2_STRFILTER: e bin.strfilter # r2 -qe bin.strfilter=? -c '' --
RABIN2_STRPURGE: e bin.strpurge # try to purge false positives
RABIN2_DEBASE64: e bin.debase64 # try to debase64 all strings
RABIN2_DMNGLRCMD: e bin.demanglercmd # try to purge false positives
RABIN2_PDBSERVER: e pdb.server # use alternative PDB server
RABIN2_PREFIX: e bin.prefix # prefix symbols/sections/relocs with a specific string
esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/usr/sbin$ rabin2 -l dropbear
Warning: Cannot initialize section headers
Warning: Cannot initialize strings table
Warning: Cannot initialize dynamic strings
[Linked libraries]
libutil.so.0
libcrypt.so.0
libgcc_s.so.1
libc.so.0

4 libraries
esslp@ubuntu: ~/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/usr/sbin$

```

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-e0KIKb: 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-Aw4qxA: 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**

```

Binary file bin/gzip matches
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
Binary file usr/lib/luajit/luajit.so matches

```

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:

```

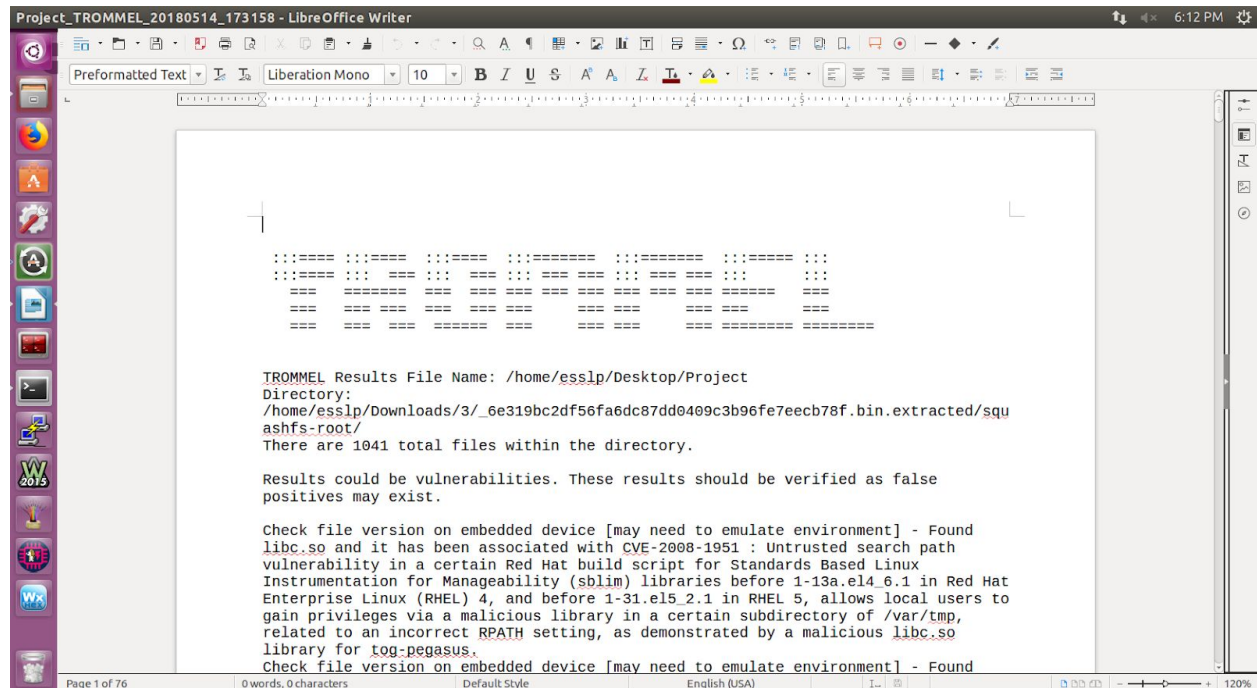
esslp@ubuntu: ~/workspace/embedtools/trommel
(esslp@ubuntu) esslp@ubuntu:~/workspace/embedtools$ cd trommel/
(esslp@ubuntu) esslp@ubuntu:~/workspace/embedtools/trommel$ ~/workspace/embedtools/trommel/trommel.py -p /home/esslp/Downloads/3/3.zip
(esslp@ubuntu) esslp@ubuntu:~/workspace/embedtools/trommel$ ~/workspace/embedtools/trommel/trommel.py -p /home/esslp/Downloads/3/_6e319bc2df56fa6dc87dd0409c3b96fe7eecb78f.bin.extracted/squashfs-root/ -o /home/esslp/Desktop/ProjectDB: /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'

(esslp@ubuntu) 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>