The Quest For Kali Nethunter : HTC One

Part 1

Attempt to port Kali Nethunter to the HTC One X.

https://www.offensive-security.com/kali-linux-nethunter-download/

The stages involved are :-

1. Build CM11 Rom
2. Modify Android source code with Nethunter kernel patches
3. Build rom with patches
4. Install Nethunter

WARNING: Be comfortable with flashing roms/ recovery images to your phone as well as using adb, fastboot and recovery. You canbrick your phone.

Pre-requesites:-

- HTC One X (endeavoru) with unlocked bootloader with S-on or S-off and TWRP Recovery 2.7.0.0
- Kali Linux 64 bit OS with terminal set for infinite scrolling
- Sun Java 1.7.0.17
- Working Android SDK in your path

First we install dependancies.

```
sudo apt-get install bison build-essential curl flex git gnupg gperf libesd0-dev libncurses5-dev libsdl1.2-dev libwxgtk2.8-dev libxml2 libxml2-utils lzop openjdk-6-jdk openjdk-6-jre pngcrush schedtool squashfs-tools xsltproc zip zlib1g-dev g++-multilib gcc-multilib lib32ncurses5-dev lib32readline-gplv2-dev lib32z1-dev
```

Next we set up the Android Build system.

```
$ mkdir -p ~/bin
$ mkdir -p ~/android/system
$ export PATH=/root/bin:$PATH
$ cd ~/android/system/
$ repo init -u https://github.com/CyanogenMod/android.git -b cm-11.0
```

This will take some time to pull the Android source code tree.Once finished the next stages are to get the prebuilt stuff and the proprietary binaries from your phone. If you do not extract the correct files into the build system then the rom will not function correctly. You should make sure that all files are copied.

```
$ cd ~/android/system/vendor/cm
$ ./get-prebuilts
<pre>
```

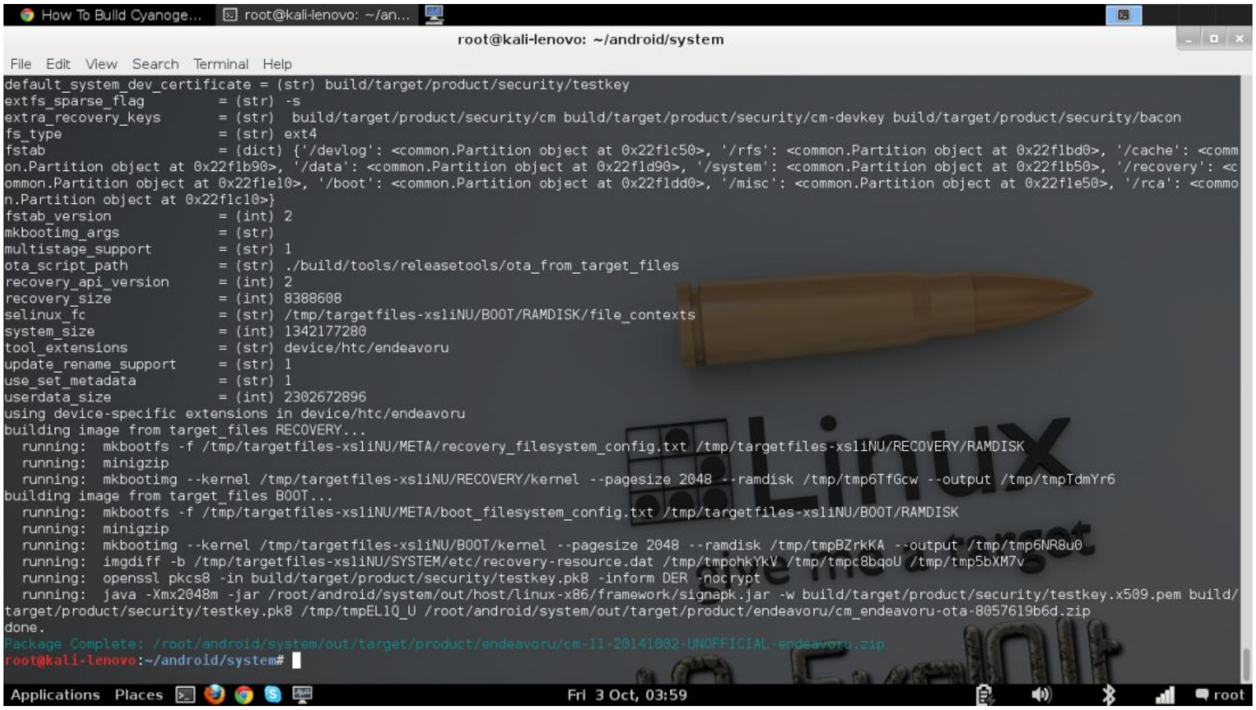
Connect your phone via usb and check the phone is recognised via ADB before downloading proprietary binaries.

```
$ adb start-server
$ adb devices
$ cd ~/android/system/device/htc/endeavoru
$ ./extract-files.sh
```

Now ,we change to the build directory and set up the compiler cache to speed up the build process. The cache is set to 50 Gigabytes.

```
$ cd ~/android/system/
$ export USE_CCACHE=1
$ prebuilts/misc/linux-x86/ccache/ccache -M 50G
$ breakfast - choose option 25 - cm_endeavoru_userdebug
$ brunch - choose option 25 - cm_endeavoru_userdebug
```

You should have a successful compile as shown by the image below:-


How To Build Cyanoge...
root@kali-lenovo: ~/an...
root@kali-lenovo: ~/android/system
File Edit View Search Terminal Help
Applications Places
Fri 3 Oct, 03:59
root

```
default_system_dev_certificate = (str) build/target/product/security/testkey
extfs_sparse_flag         = (str) -s
extra_recovery_keys       = (str)  build/target/product/security/cm build/target/product/security/cm-devkey build/target/product/security/bacon
fs_type                   = (str) ext4
fstab                     = (dict) {'/devlog': <common.Partition object at 0x22f1c50>, '/rfs': <common.Partition object at 0x22f1bd0>, '/cache': <comm
on.Partition object at 0x22f1b90>, '/data': <common.Partition object at 0x22f1d90>, '/system': <common.Partition object at 0x22f1b50>, '/recovery': <c
ommon.Partition object at 0x22f1e10>, '/boot': <common.Partition object at 0x22f1dd0>, '/misc': <common.Partition object at 0x22f1e50>, '/rca': <commo
n.Partition object at 0x22f1c10>}
fstab_version             = (int) 2
mkbootimg_args            = (str)
multistage_support        = (str) 1
ota_script_path           = (str) ./build/tools/releasetools/ota_from_target_files
recovery_api_version      = (int) 2
recovery_size             = (int) 8388608
selinux_fc                = (str) /tmp/targetfiles-xsliNU/BOOT/RAMDISK/file_contexts
system_size               = (int) 1342177280
tool_extensions           = (str) device/htc/endeavoru
update_rename_support     = (str) 1
use_set_metadata          = (str) 1
userdata_size             = (int) 2302672896
using device-specific extensions in device/htc/endeavoru
building image from target_files RECOVERY...
  running:  mkbootfs -f /tmp/targetfiles-xsliNU/META/recovery_filesystem_config.txt /tmp/targetfiles-xsliNU/RECOVERY/RAMDISK
  running:  minigzip
  running:  mkbootimg --kernel /tmp/targetfiles-xsliNU/RECOVERY/kernel --pagesize 2048 --ramdisk /tmp/tmp6TfGcw --output /tmp/tmpTdmYr6
building image from target_files BOOT...
  running:  mkbootfs -f /tmp/targetfiles-xsliNU/META/boot_filesystem_config.txt /tmp/targetfiles-xsliNU/BOOT/RAMDISK
  running:  minigzip
  running:  mkbootimg --kernel /tmp/targetfiles-xsliNU/BOOT/kernel --pagesize 2048 --ramdisk /tmp/tmpBZrkKA --output /tmp/tmp6NR8u0
  running:  imgdiff -b /tmp/targetfiles-xsliNU/SYSTEM/etc/recovery-resource.dat /tmp/tmpohkYkV /tmp/tmpc8bqoU /tmp/tmp5bXM7v
  running:  openssl pkcs8 -in build/target/product/security/testkey.pk8 -inform DER -nocrypt
  running:  java -Xmx2048m -jar /root/android/system/out/host/linux-x86/framework/signapk.jar -w build/target/product/security/testkey.x509.pem build/
target/product/security/testkey.pk8 /tmp/tmpEL1Q_U /root/android/system/out/target/product/endeavoru/cm_endeavoru-ota-8057619b6d.zip
done.
Package Complete: /root/android/system/out/target/product/endeavoru/cm-11-20141002-UNOFFICIAL-endeavoru.zip
root@kali-lenovo:~/android/system#
```

Part 2

In this post, we shall look at flashing the rom that we built in Part 1. Firstly, my phone has an unlocked bootloader, which is needed to flash custom roms. If your bootloader is locked head over to thehtcdev.com and click on the icon marked “Unlock Bootloader”. Once you have unlocked your bootloader, your phone may still have S-ON enabled, this is a security measure which prevents flashing of the boot partition on the phone. It can be resolved by manually flashing the boot.img file using fastboot.

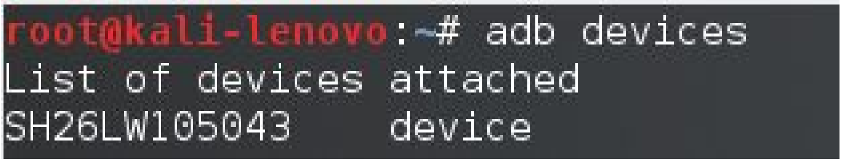
The flashing process follows the following steps

start adb server and check phone is connected

1. use adb push to push all files to required places
2. reboot phone into bootloader mode
3. put phone in fast boot mode
4. flash boot.img file using fastboot
5. reboot the phone into recovery
6. perform a wipe
7. install from zip, the image you transfered
8. reboot phone

WARNING: You should be comfortable with flashing roms/ recovery images to your phone as well as using adb, fastboot and recovery. I am not responsible if you brick your phone.

First make sure that your phone is connected as shown below



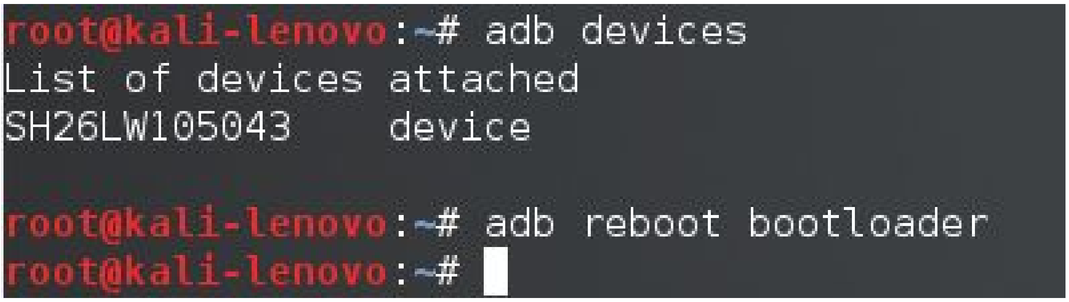
```
root@kali-lenovo:~# adb devices
List of devices attached
SH26LW105043    device
```

Now from the terminal type the following

```
cd /root/android/system/out/target/product/endeavoru
adb push ./cm-11-20141009-UNOFFICIAL-endeavoru.zip /sdcard/Download/cm-11-rom.zip
```

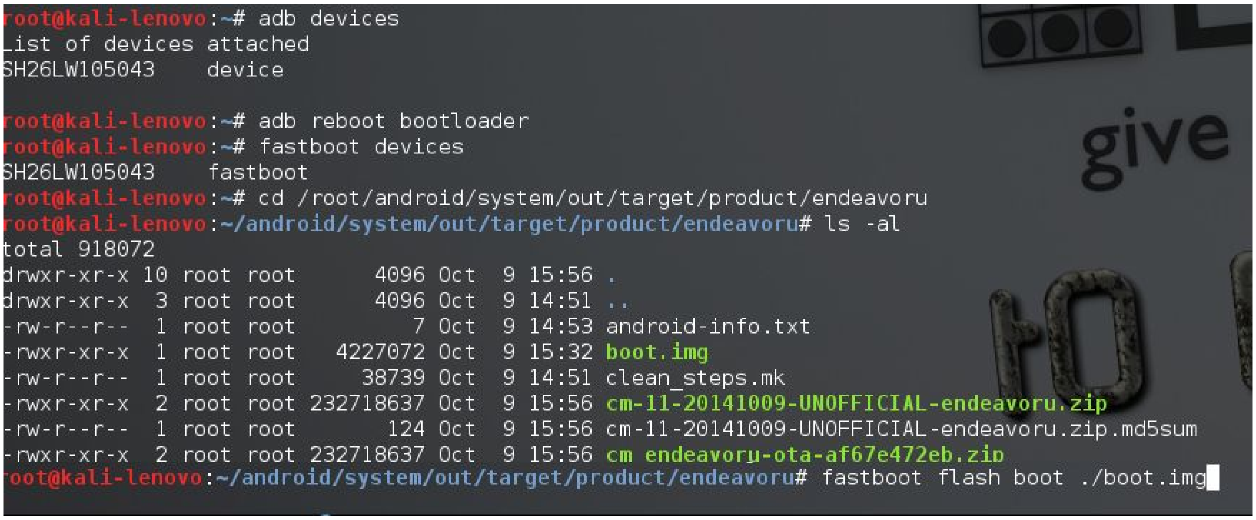
Note: your zip file name will be different

Now reboot into ‘bootloader’ mode.



```
root@kali-lenovo:~# adb devices
List of devices attached
SH26LW105043    device

root@kali-lenovo:~# adb reboot bootloader
root@kali-lenovo:~# 
```

Now on the handset , select “fast boot usb” mode. Make sure the device is connected properly, change to the ‘out’ directory and change the permissions to 755 on the ‘boot.img’ file as shown below. Now simply enter the fastboot flash command as shown.



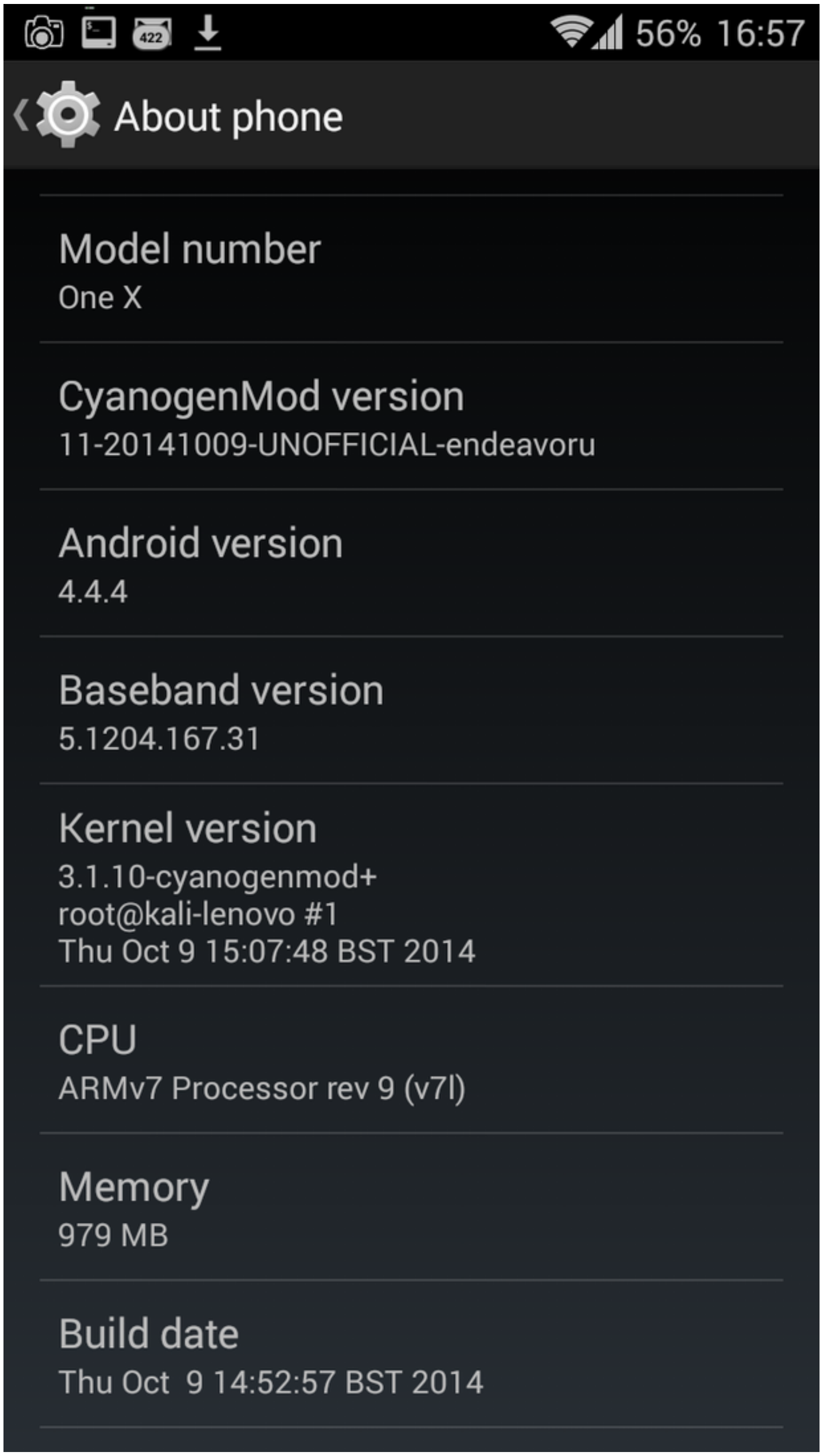
```
root@kali-lenovo:~# adb devices
List of devices attached
SH26LW105043    device

root@kali-lenovo:~# adb reboot bootloader
root@kali-lenovo:~# fastboot devices
SH26LW105043    fastboot
root@kali-lenovo:~# cd /root/android/system/out/target/product/endeavoru
root@kali-lenovo:~/android/system/out/target/product/endeavoru# ls -al
total 918072
drwxr-xr-x 10 root root      4096 Oct  9 15:56 .
drwxr-xr-x  3 root root      4096 Oct  9 14:51 ..
-rw-r--r--  1 root root         7 Oct  9 14:53 android-info.txt
-rwxr-xr-x  1 root root   4227072 Oct  9 15:32 boot.img
-rw-r--r--  1 root root     38739 Oct  9 14:51 clean_steps.mk
-rwxr-xr-x  2 root root 232718637 Oct  9 15:56 cm-11-20141009-UNOFFICIAL-endeavoru.zip
-rw-r--r--  1 root root       124 Oct  9 15:56 cm-11-20141009-UNOFFICIAL-endeavoru.zip.md5sum
-rwxr-xr-x  2 root root 232718637 Oct  9 15:56 cm endeavoru-ota-af67e472eb.zip
root@kali-lenovo:~/android/system/out/target/product/endeavoru# fastboot flash boot ./boot.img
```

Congratulations, you now have flashed your bootloader.

Next, reboot into recovery on the phone using the volume up/down keys and power button. Once in recovery, perform a wipe and install the cm-11-rom.zip file located in /sdcard/Downloads directory.

Reboot your phone into your android. Be patient with the boot process as it can take some time for a first boot.

If all goes well, you will have a running Android operating system as shown below.



56% 16:57

About phone

Model number
One X

CyanogenMod version
11-20141009-UNOFFICIAL-endeavoru

Android version
4.4.4

Baseband version
5.1204.167.31

Kernel version
3.1.10-cyanogenmod+
root@kali-lenovo #1
Thu Oct 9 15:07:48 BST 2014

CPU
ARMv7 Processor rev 9 (v7l)

Memory
979 MB

Build date
Thu Oct 9 14:52:57 BST 2014

In Part 3, we will modify the kernel to incorporate the patches necessary for Kali Nethunter.

Part 3

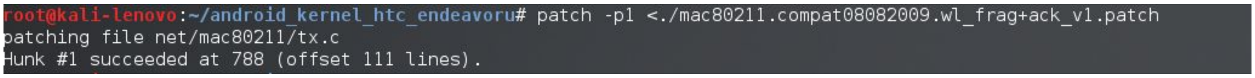
Patching of the kernel sources for cm-11.

The kernel sources is located at /root/android/system/kernel// which in my case is root/android/system/kernel/htc/endeavoru.

Next we change to kernel source tree root location and get patches from the following

wget http://patches.aircrack-ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch

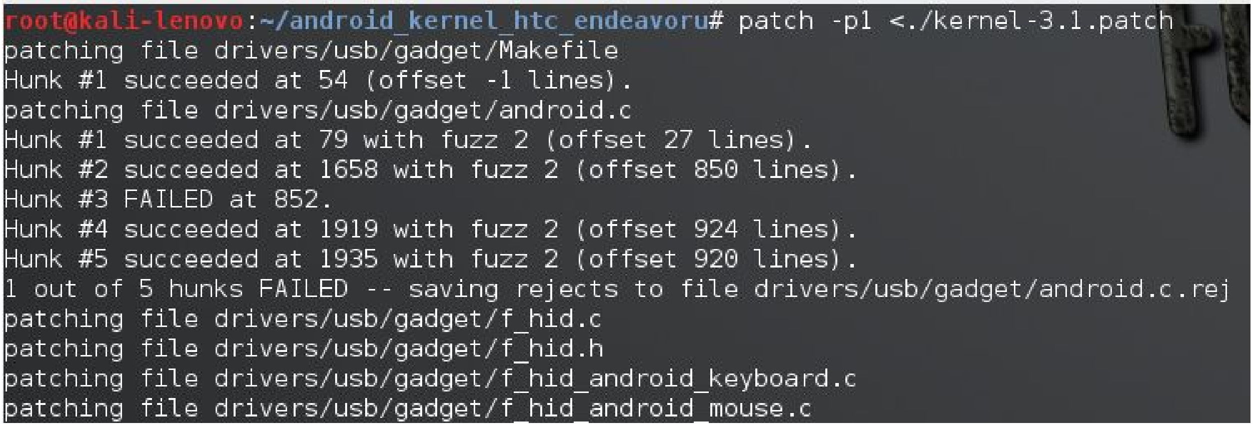
Now, we patch it.



```
root@kali-lenovo:~/android_kernel_htc_endeavoru# patch -p1 <./mac80211.compat08082009.wl_frag+ack_v1.patch
patching file net/mac80211/tx.c
Hunk #1 succeeded at 788 (offset 111 lines).
```

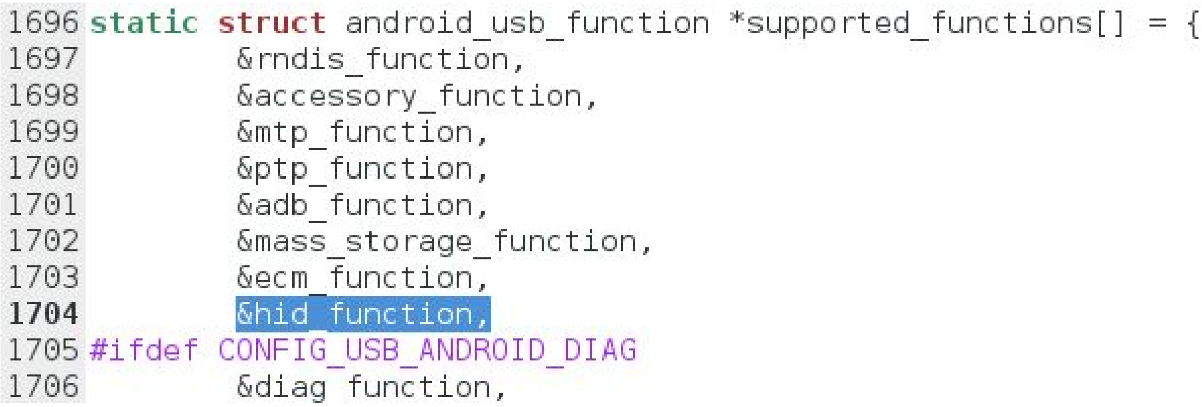
Now we download the kernel patch from pelya repository at https://github.com/pelya/android-keyboard-gadget/blob/master/kernel-3.1.patch

Next we patch it as shown below:-



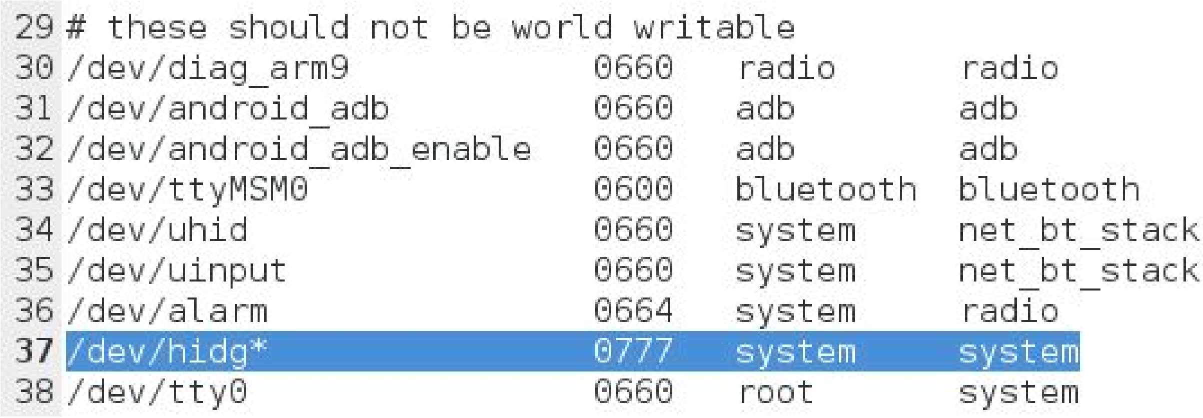
```
root@kali-lenovo:~/android_kernel_htc_endeavoru# patch -p1 <./kernel-3.1.patch
patching file drivers/usb/gadget/Makefile
Hunk #1 succeeded at 54 (offset -1 lines).
patching file drivers/usb/gadget/android.c
Hunk #1 succeeded at 79 with fuzz 2 (offset 27 lines).
Hunk #2 succeeded at 1658 with fuzz 2 (offset 850 lines).
Hunk #3 FAILED at 852.
Hunk #4 succeeded at 1919 with fuzz 2 (offset 924 lines).
Hunk #5 succeeded at 1935 with fuzz 2 (offset 920 lines).
1 out of 5 hunks FAILED -- saving rejects to file drivers/usb/gadget/android.c.rej
patching file drivers/usb/gadget/f_hid.c
patching file drivers/usb/gadget/f_hid.h
patching file drivers/usb/gadget/f_hid_android_keyboard.c
patching file drivers/usb/gadget/f_hid_android_mouse.c
```

As we can see the patching worked apart from android.c in drivers/usb/gadget . Upon looking at the code, it became clear that it could be manually patched in android.c at line 1704 as shown below:-



```
static struct android_usb_function *supported_functions[] = {
        &rndis_function,
        &accessory_function,
        &mtp_function,
        &ptp_function,
        &adb_function,
        &mass_storage_function,
        &ecm_function,
        &hid_function,
#ifdef CONFIG_USB_ANDROID_DIAG
        &diag_function,
```

Now that we have patched the kernel, we need to apply the eventdrc patch in /root/android/system/system/core/rootdir as shown below:-



```
# these should not be world writable
/dev/diag_arm9            0660   radio       radio
/dev/android_adb          0660   adb         adb
/dev/android_adb_enable   0660   adb         adb
/dev/ttyMSM0              0600   bluetooth   bluetooth
/dev/uhid                 0660   system      net_bt_stack
/dev/uinput               0660   system      net_bt_stack
/dev/alarm                0664   system      radio
/dev/hidg*                0777   system      system
/dev/tty0                 0660   root        system
```

Now head to the android/system and type ‘make clean’ to clean the build system. Now rebuild the rom.

Attributions:

- Cyberkryption - **Paul Dutot** - **Twitter** - @pauldutot
  https://twitter.com/pauldutot
- **Gennaro Cimmino -Git Hub** l0rdg3x
  https://github.com/l0rdg3x
- Quest: Culminated by Frank Beatrice **- Git Hub - FrankBeatrice**
  https://github.com/FrankBeatrice