**Necessary Resources:**

Rooted Nexus 5 Phone

TP-LINK AC1750 router

10GB Storage Space

**‘Root Nexus 5 (This is done on our Nexus5 phone, do not repeat):**

[https://www.online-tech-tips.com/smartphones/how-to-root-a-nexus-5-5x-6-6p-7/#:~:text=Press%20and%20hold%20down%20Volume,it'll%20root%20your%20device](https://www.online-tech-tips.com/smartphones/how-to-root-a-nexus-5-5x-6-6p-7/).

**Nexmon Installation guide:**

[https://github.com/seemoo-lab/nexmon\_csi#getting-started](https://github.com/seemoo-lab/nexmon_csi)

* Use on Ubuntu 18.04 (Bionic Beaver) or *22.04 (Jammy Jellyfish) Recommended, necessary for live script*

Installation:

0. Find a VirtualBox instance with sudo privileges, or follow the below tutorial to get them: <https://phoenixnap.com/kb/how-to-create-sudo-user-on-ubuntu>, virtualbox download <https://www.virtualbox.org/wiki/Downloads>

sudo usermod -aG sudo vboxuser

groups vboxuser

You should see vboxuser : vboxuser sudo

1. Install some dependencies: sudo apt-get install git gawk qpdf adb flex bison

2. Only necessary for x86\_64 systems, install i386 libs:

$ sudo dpkg --add-architecture i386

$ sudo apt-get update

$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386

Go into the home directory.

3. Clone the nexmon base repository: **$ git clone** [**https://github.com/seemoo-lab/nexmon.git**](https://github.com/seemoo-lab/nexmon.git).

4. Download and extract Android NDK r11c Linux (use exactly this version!).

Link: <https://github.com/android/ndk/wiki/Unsupported-Downloads>

Note - Move the only folder within that one which was installed to the home directory.

Step 4: Before continuing, type **$ sudo bash**

5. Export the NDK\_ROOT environment variable pointing to the location where you extracted the ndk so that it can be found by our build environment.

Step 5: for Xubuntu

Add these two lines to setup\_env.sh

**$ export NDK\_ROOT=/<insert path to extracted folder>/android-ndk-r11c**

**$ export PATH=$NDK\_ROOT:$PATH**

Test with **$ echo $NDK\_ROOT**

Note: Make sure the environment **gcc, g++ and python** is installed.!!! If not, run:

**$ sudo apt-get install gcc**

**$ sudo apt-get install g++**

**$ sudo apt-get install python** (\*\*if in Jammy Jellyfish do **$ sudo apt-get install python2**, then go to /usr/bin/python2 and put in **$ mv python2 python**\*\*)

6. Navigate to the previously cloned nexmon directory and execute

$ **source setup\_env.sh**

to set a couple of environment variables.

7. Run $ **make** to extract ucode, templateram and flashpatches from the original firmwares.

8. Navigate to utilities and run $ **make** to build all utilities such as nexmon.

Note - utilities is in Nexmon, not a default folder

Check nested folder issues

**Step 8: Takes a long time**

9. Attach your rooted Nexus 5 or Nexus 6P smartphone.

10. Run $ **make install** to install all the built utilities on your phone.

11. Navigate to patches/bcm4339/6\_37\_34\_43/ and clone this repository:

$ **git clone** [**https://github.com/seemoo-lab/nexmon\_csi.git**](https://github.com/seemoo-lab/nexmon_csi.git)

12. Enter the created subdirectory nexmon\_csi and run

$ **make install-firmware**

to compile our firmware patch and install it on the attached smartphone.

If you have already installed, still execute these steps before usage:

5. Export the NDK\_ROOT environment variable pointing to the location where you extracted the ndk so that it can be found by our build environment.

Step 5: for Ubuntu

**export NDK\_ROOT=/<insert path to extracted folder>/android-ndk-r11c**

**export PATH=$NDK\_ROOT:$PATH**

Step 5: make sure the environment variable is set, otherwise following steps will result in error

6. Navigate to the previously cloned nexmon directory and execute $ source setup\_env.sh to set a couple of environment variables.

7. Run **$ make** to extract ucode, templateram and flashpatches from the original firmwares.

8. Navigate to utilities and run **$ make** to build all utilities such as nexmon.

Usage:

Make sure device is mounted onto Ubuntu: Devices -> USB -> Nexus 5

0. Go to patches/bcm4339/6\_37\_34\_43/nexmon\_csi/utils/makecsiparams and make.

1. Use utils/makecsiparams/makecsiparams to generate a base64 encoded parameter string that can be used to configure the extractor. The following example call generates a parameter string that enables collection on channel 157 with 80 MHz bandwidth on the first core for the first spatial stream for frames starting with 0x88 originating from 00:11:22:33:44:55 or aa:bb:aa:bb:aa:bb:

**$ ./makecsiparams -c 153/80 -C 1 -N 1 -m 98:DA:C4:8E:4D:8E**

**$ ./makecsiparams -c 153/20 -C 1 -N 1 -m 98:DA:C4:8E:4D:8E**

in our case, parameter string = m+EBEQAAAQCY2sSOTY4AAAAAAAAAAAAAAAAAAAAAAAAAAA==

or

mdABEQAAAQCY2sSOTY4AAAAAAAAAAAAAAAAAAAAAAAAAAA==

Or

m+IBEQGIAgAAESIzRFWqu6q7qrsAAAAAAAAAAAAAAAAAAA==

For a full list of possible parameters run

$ ./makecsiparams -h

2. Run adb shell and then su to get into the phone command line.

3. Make sure your interface is up: $ **ifconfig wlan0 up**

4. Configure the extractor using nexutil and the generated parameters:

**$ nexutil -Iwlan0 -s500 -b -l34 -v m+EBEQAAAQCY2sSOTY4AAAAAAAAAAAAAAAAAAAAAAAAAAA==**

**Or**

**$ nexutil -Iwlan0 -s500 -b -l34 -v mdABEQAAAQCY2sSOTY4AAAAAAAAAAAAAAAAAAAAAAAAAAA==**

Or

**$ nexutil -Iwlan0 -s500 -b -l34 -v m+IBEQGIAgAAESIzRFWqu6q7qrsAAAAAAAAAAAAAAAAAAA==**

5. Enable monitor mode:

**$ nexutil -Iwlan0 -m1**

6. Collect CSI by listening on UDP socket 5500, e.g. by using tcpdump:

**$ tcpdump -i wlan0 -c 50 -vv dst port 5500 -w /sdcard/csi.pcap**

There will be one UDP packet per configured core and spatial stream for each incoming frame matching the configured filter.

Change the value after -c to change the number of packets recieved.

FROM DESKTOP:

7. do $ **adb pull /sdcard/csi.pcap ~/Desktop/csi.pcap** to put the file on your computer

**To find the MAC address of the target router: type**

**$ iwlist wlan0 scan | grep -C 5 "TP-Link\_4D8F\_5G" <----- The name of the router**

Step 3/4 Notes: If phone disconnects from wifi, type

**$ ifconfig wlan0 down**

then

**$ ifconfig wlan0 up**

to reconnect

For Live CSI Read (Must be on Jammy Jellyfish - see note above about renaming python2 -> python):

1. Install some (more) dependencies, after installing pip3:
   1. $ pip3 install tensorflow
   2. $ pip3 install csiread
   3. $ pip3 install matplotlib
2. Download the csi\_lv folder in the Google Drive
3. Enter the folder in terminal and enter **$ bash run\_lv.sh 1** to execute