The purpose of this document.

To provide a step by step guide to build a Rasberry Pi Zero W with HCXDumpTool. Installed and functional using the onboard Broadcom bcm43430 WIFI chip on Raspian Stretch lite OS

Download and install the Raspian Stretch Light image located at;

<https://www.raspberrypi.org/downloads/raspbian/>

install image on high speed SD card

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

In /boot/config.txt located on the SD card add enable\_uart=1 on the last line and save.

<https://learn.adafruit.com/adafruit-piuart-usb-console-and-power-add-on-for-raspberry-pi/enabling-serial-console>

Place the SD card into your PI and make sure you have the PI connected to your PC.

Place example pic here

* Boot PI and Login
* U = pi
* P = raspberry
* Elevate Permissions
* sudo su
* Install PreRecs
* apt install git libpcap-dev libcurl4-openssl-dev libssl-dev libwiringpi wiringpi
* Install libpcap
* <http://www.linuxfromscratch.org/blfs/view/svn/basicnet/libpcap.html>
* wget <http://www.tcpdump.org/release/libpcap-1.8.1.tar.gz>
* Download NEXMON
* cd /opt
* git clone https://github.com/seemoo-lab/nexmon.git

Build patches for bcm43430a1 on the RPI3/Zero W using Raspbian Stretch

<https://github.com/seemoo-lab/nexmon#build-patches-for-bcm43430a1-on-the-rpi3zero-w-using-raspbian-stretch-recommended>

**Note:** We currently support Kernel Version 4.4 (depricated) and 4.9

* Make sure the following commands are executed as root:
* sudo su
* Upgrade your Raspbian OS installation:
* apt-get update && apt-get upgrade
* Install the kernel headers to build the driver and some dependencies:
  + sudo apt install raspberrypi-kernel-headers libgmp3-dev gawk qpdf bison flex make git tcpdump libpthread-stubs0-dev
  + cd /opt/
  + git clone git://git.drogon.net/wiringPi
  + cd ~/wiringPi
  + $ ./build
* Clone our repository:
  + git clone https://github.com/seemoo-lab/nexmon.git
* Go into the root directory of our repository:
  + cd nexmon
* Check if /usr/lib/arm-linux-gnueabihf/libisl.so.10 exists
  + ls /usr/lib/arm-linux-gnueabihf/libisl.so.10
* if not, compile it from source, install and create symLink:
  + cd buildtools/isl-0.10
  + ./configure
  + make
  + make install
  + ln -s /usr/local/lib/libisl.so /usr/lib/arm-linux-gnueabihf/libisl.so.10

Setup the build environment for compiling firmware patches

* Return to the nexmon root dir and setup the build environment:
* cd ../../
* source setup\_env.sh
* Compile some build tools and extract the ucode and flashpatches from the original firmware files:
* make
* Go to the *patches* folder for the bcm43430a1 chipset:
* cd patches/bcm43430a1/7\_45\_41\_46/nexmon/
* Compile a patched firmware:
* make
* Generate a backup of your original firmware file:
* make backup-firmware
* Install the patched firmware:
* make install-firmware

Install nexutil:

* From the root directory of our repository switch to the nexutil folder:
* cd nexmon/utilities/nexutil/.
* Compile and install nexutil:
* make && make install.
* *Optional*: remove wpa\_supplicant for better control over the WiFi interface. **NOTE:**I have found that I am unable to get monitor mode working if wpa supplicant is installed
* apt-get remove wpasupplicant

**Note:** To connect to regular access points you have to execute the following command first:

* nexutil -m0

*Optional*: To make the RPI3 load the modified driver after reboot:

* Find the path of the default driver at reboot:
* modinfo brcmfmac | grep brcmfmac.ko
* Backup the original driver:
* mv "<PATH TO THE DRIVER>/brcmfmac.ko" "<PATH TO THE DRIVER>/brcmfmac.ko.orig"
* Copy the modified driver:
* cp /opt/nexmon/patches/bcm43430a1/7\_45\_41\_26/nexmon/brcmfmac\_kernel49/brcmfmac.ko /lib/modules/4.14.62+/kernel/drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.ko
* Probe all modules and generate new dependency:
* depmod -a
* The new driver should be loaded by default after reboot:
* reboot

### Using the Monitor Mode patch

* Thanks to the prior work of Mame82, you can setup a new monitor mode interface by executing:
  + iw phy `iw dev wlan0 info | gawk '/wiphy/ {printf "phy" $2}'` interface add mon0 type monitor
* To activate monitor mode in the firmware, simply set the interface up:
  + ifconfig mon0 up.
* At this point, monitor mode is active. There is no need to call airmon-ng.
* The interface already set the Radiotap header, therefore, tools like tcpdump or airodump-ng can be used out of the box:
  + tcpdump -i mon0

Optional: To make the RPI3 load the modified driver after reboot:

* + Find the path of the default driver at reboot: modinfo brcmfmac #the first line should be the full path
  + Backup the original driver: mv "<PATH TO THE DRIVER>/brcmfmac.ko" "<PATH TO THE DRIVER>/brcmfmac.ko.orig"
  + Get Kernel version info: uname -a
  + Copy the modified driver for (Kernel 4.9): cp /home/pi/nexmon/patches/bcm43430a1/7\_45\_41\_46/nexmon/brcmfmac\_kernel49/brcmfmac.ko "<PATH TO THE DRIVER>/"
  + Copy the modified driver (Kernel 4.14): cp /opt/nexmon/patches/bcm43430a1/7\_45\_41\_46/nexmon/brcmfmac\_4.14.y-nexmon/brcmfmac.ko "<PATH TO THE DRIVER>/"
  + Probe all modules and generate new dependency: depmod -a
  + The new driver should be loaded by default after reboot: reboot  \* **Note:** It is possible to connect to an access point or run your own access point in parallel to the monitor mode interface on the wlan0 interface
* I have found that I have to run these two commands every time I want to run monitor mode so, I have added the following two lines to /etc/rc.local script at the bottom before exit 0
  + iw phy `iw dev wlan0 info | gawk '/wiphy/ {printf "phy" $2}'` interface add mon0 type monitor
  + ifconfig mon0 up
  + Save and reboot

**Example of rc.local edits:**

GNU nano 2.7.4 File: etc/rc.local Modified

#

# By default this script does nothing.

# Print the IP address

\_IP=$(hostname -I) || true

if [ "$\_IP" ]; then

printf "My IP address is %s\n" "$\_IP"

fi

iw phy `iw dev wlan0 info | gawk '/wiphy/ {printf "phy" $2}'` interface add mon$

ifconfig mon0 up

exit 0

At this point, monitor mode is active on system startup. There is no need to call *airmon-ng*.

* The interface has already set the Radiotap header, therefore, tools like *tcpdump* or *airodump-ng* can be used out of the box:
  + tcpdump -i mon0 ( confirm Monitor mode is working then cancel)

Install HCX Utilz by ZERBEA

hcxdumptool

* + git clone <https://github.com/ZerBea/hcxdumptool.git>
* Compile hcxdumptool
  + cd hcxdumptool
  + Make
  + make install (as super user)
* or (with GPIO support - hardware mods required)
  + make GPIOSUPPORT=on
  + make GPIOSUPPORT=on install (as super user)