Capturing Wireless LAN Packets in Monitor Mode with iw

Originalartikel

Backup

<html> I previously showed two ways to capture wireless LAN packets in Ubuntu Linux: using the command line tool <var class=„cmd“>iwconfig</var> and using Kismet. Both involve putting the wireless LAN card into „monitor mode“, allowing you to view and record all packets sent by other WiFi devices nearby. This includes data packets send between other devices, something which is not possible unless your device is in monitor mode. Here I present a third option: again using the command line in Ubuntu Linux but with the command <var class=„cmd“>iw</var>. The command iw is meant to replace iwconfig. I still like and use the old interface of iwconfig, but iw seems to be much more powerful for viewing/configuring wireless information. <h3>Getting Started with iw</h3> First be aware that <var class=„cmd“>iw</var> distinguishes between wireless LAN hardware devices (the physical layer, referred to as <tt>phy</tt>) and the network interface configured to use that hardware (e.g. <tt>wlan0</tt>, similar to an Ethernet <tt>eth0</tt> interface). To see the list of devices, and interfaces for each device: <pre>$ <kbd>iw dev</kbd> phy#0

```
	Interface wlan0
		ifindex 3
		type managed
```

</pre> In my case (and most likely for most typical computers) the hardware is <var class=„addr“>phy0</var> and my network interface is <var class=„addr“>wlan0</var>. You can see detailed information about the hardware using: <pre>$ <kbd>iw phy phy0 info</kbd> Wiphy phy0

```
	Band 1:
		Capabilities: 0x172
			HT20/HT40
```

…

```
	Supported interface modes:
		 * IBSS
		 * managed
		 * AP
		 * AP/VLAN
		 * WDS
		 * monitor
		 * mesh point
	software interface modes (can always be added):
		 * AP/VLAN
		 * monitor
```

… </pre> Of importance for the next step is the supported/software interface modes should include entry for „monitor“, meaning your hardware supports monitor mode. If there is no „monitor“ entry, then you will not be able to capture other peoples data using the next steps. <h3>Capturing in Monitor Mode</h3> If your hardware device supports monitor mode then you must add a monitor interface called <var class=„addr“>mon0</var>. <pre>$ <kbd>sudo iw phy phy0 interface add mon0 type monitor</kbd> </pre> You can check that it is added: <pre>$ <kbd>iw dev</kbd> phy#0

```
	Interface mon0
		ifindex 4
		type monitor
	Interface wlan0
		ifindex 3
		type managed
```

</pre> We will capture with the <var class=„addr“>mon0</var> interface, so you can delete the normal <var class=„addr“>wlan0</var> interface: <pre>$ <kbd>sudo iw dev wlan0 del</kbd> </pre> Now enable the <var class=„addr“>mon0</var> interface using <var class=„cmd“>ifconfig</var>: <pre>$ <kbd>sudo ifconfig mon0 up</kbd> </pre> Before capturing, specify the wireless LAN frequency you want to capture on. You should choose the frequency based on the channels used by neighbouring access points. The frequency is given in MHz, e.g.\ channel 6 is <tt>2437</tt>. <figure><figcaption>2.4 GHz Wi-Fi channels (802.11b,g WLAN), Michael Gauthier / Wikimedia Commons / CC-BY-SA-3.0 /</figcaption></figure><pre>$ <kbd>sudo iw dev mon0 set freq 2437</kbd> </pre> To check that your interface is in monitor mode and using the correct frequency you can use <var class=„cmd“>iwconfig</var>: <pre>$ <kbd>iwconfig mon0</kbd> mon0 IEEE 802.11bgn Mode:Monitor Frequency:2.437 GHz Tx-Power=20 dBm

```
	Retry  long limit:7   RTS thr:off   Fragment thr:off
	Power Management:on
```

</pre> Now you can capture, e.g. using <var class=„cmd“>tcpdump</var>: <pre>$ <kbd>sudo tcpdump -i mon0 -n -w wireless.cap </kbd></pre> <kbd>Ctrl-C to stop the capture, then view with Wireshark. To display select wireless LAN frames in Wireshark use the wlan and wlan_mgt filters. (My brief summary of Wireshark and WLAN filters)</kbd> <h3><kbd>Returning to Managed Mode</kbd></h3> <kbd>If after monitoring you want to revert the changes and continue using the <var class=„addr“>wlan0</var> interface in managed mode (e.g. connect to an AP), then delete the <var class=„addr“>mon0</var> interface and add the <var class=„addr“>wlan0</var> interface:</kbd> <pre><kbd>

```
$ <kbd>sudo iw dev mon0 del</kbd>
$ <kbd>sudo iw phy phy0 interface add wlan0 type managed</kbd>
$ <kbd>iw dev</kbd>
```

```
phy#0
        Interface wlan0
                ifindex 5
                type managed
$ <kbd>iwconfig wlan0</kbd>
wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
```

</kbd></pre> <h3>What Can Go Wrong?</h3> If you get errors with some of the above <var class=„cmd“>iw</var> commands, then: <ol readability=„0“><li readability=„1“>Check that the wireless device is not soft/hard blocked by <var class=„cmd“>rfkill</var> and unblock it if it is: <pre>$ <kbd>rfkill list</kbd> 0: phy0: Wireless LAN

```
      Soft blocked: yes
      Hard blocked: no
```

$ <kbd>rfkill unblock 0</kbd> </pre> Make sure you are using the correct interface/device. In my examples I use <var class=„addr“>phy0</var>, <var class=„addr“>wlan0</var> and <var class=„addr“>mon0</var>. Yours may be different. If the commands work, but in Wireshark you can only see packets either to your computer or broadcast/multicast (i.e. you cannot see any packets from one computer to another computer, such as HTTP or SSH), then: Make sure the frequency you selected is being used by others. Check that your device supports monitor mode (look in the output of <var class=„cmd“>iw phy phy0 info</var>). Some wireless cards do not support monitor mode, and even if they do, some drivers do not support it. <h3>Selecting a Wireless Adapter that Supports Monitor Mode</h3> On the last point above, finding a wireless adapter that supports monitor mode to allow capturing of data packets in Linux can be troublesome. It depends on both the hardware and driver support. The Linux wireless drivers page provides a quick summary of monitor mode support in different drivers. To find out which devices use which drivers you can search on WikiDevi. As of March 2015, devices that use Atheros, Intel, RaLink or Broadcom chipsets seem to have good monitor mode support. When looking to buy a wireless USB (or PCI) adapter that will support monitor mode, find some devices that are available, look them up on WikiDevi to see the drivers, and then check the driver support for monitor mode. Be especially careful with hardware versions: many branded devices are updated over time and although they have the same model number, the internal wireless chipset may change. Some devices I have successfully used include: USB (2014): generic brand, made by Shenzhen (pic), similar to Tenda W311MI but with attachable antenna, RaLink RT5370 (driver: rt2800usb) USB (2012): D-Link DWA-160 HW version A2, Atheros AR9170 (carl9170) USB (2011): Alfa Networks AWUS036H, Realtek 8187 (rtl8187) MiniPCI (2011): Intel Centrino Wireless-N 1000 (iwlwifi) MiniPCI (2009): Atheros (ath5k) PCI (2009): SMC WPCIG Atheros AR5007G (ath5k?) I recently ordered a batch of TP-Link TL-WN821N after checking that it supported monitor mode, but when delivered it was hardware version 4 which used a Realtek chipset (rtl8192cu) that did NOT fully support monitor mode (version 1 to 3 used Atheros chips, which did support monitor mode). Be carefully in checking the specific hardware versions when purchasing wireless devices. Created on Mon, 09 Mar

2015, 6:51pm Last changed on Tue, 10 Mar 2015, 8:15pm </html>