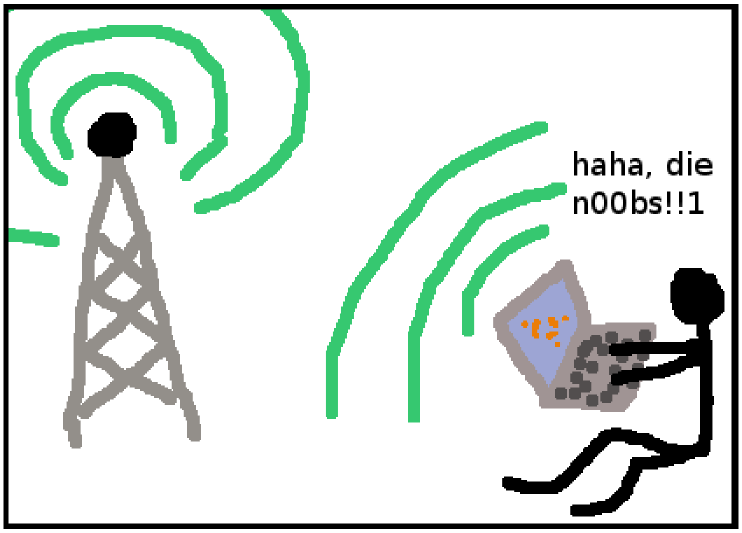
Wi-fi packet sniffing on Ubuntu

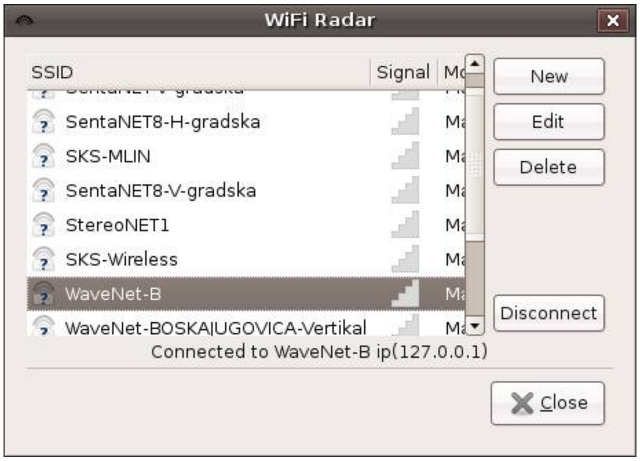


Packet sniffers let you take a look at the network traffic of a wireless network, seeing what websites are people visiting, watch chat conversations, capture some unencrypted passwords, or just see how packets flow through different protocols. This tutorial will show you how to discover wireless network and passively (without interfering) sniff their traffic on Ubuntu (Edgy).

Drivers

Although many WLAN cards are supported by default in Ubuntu, some are not.
Try downloading and installing the latest Orinoco or Prism2 drivers.
Links are listed in the end of the text.

Discovering wireless networks



SSID	Signal	Mc
SentaNET8-H-gradska		Ma
SKS-MLIN		Ma
SentaNET8-V-gradska		Ma
StereoNET1		Ma
SKS-Wireless		Ma
WaveNet-B		Ma
WaveNet-BOSKAJUGOVICA-Vertikal		Ma

Connected to WaveNet-B ip(127.0.0.1)

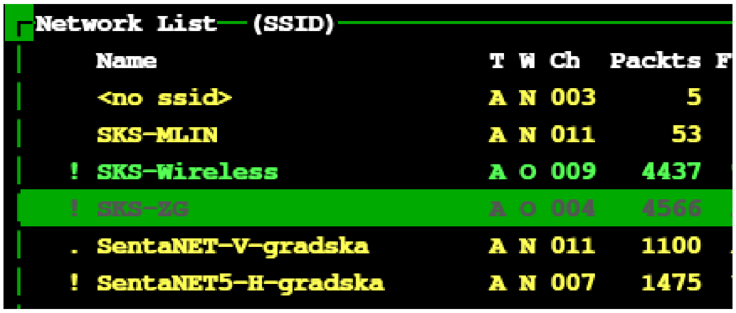
You will need a tool to discover wireless networks and get information about them.

Probably the easiest way to do this is installing a package called Wifi-radar.
Wifi-radar will let you see the SSID, signal strenght, mode and type of the 802.11 standard (b, g, etc.)
It doesn't provide much details, but for very basic use it's OK.

Of course, there are tools that provide more details about wireless networks.
Kismet is probably the most popular tool among wardrivers who use Linux.
It sniffs traffic passively, making it impossible to be detected and it even supports GPS.
So go to http://packages.ubuntu.com and download Kismet from the networking section. Install the package.
Before you start, you'll have to modify the kismet.conf file.
Open /etc/kismet/kismet.conf with your favourite text editor and modify the source option to fit your needs. For example:

```
source=orinoco,wlan0,kismet
```

Now start Kismet from the terminal, using the `sudo kismet` command (root privileges).



```
Network List── (SSID)──────────────────────────
      Name                           T W Ch  Packts F
      <no ssid>                      A N 003      5
      SKS-MLIN                       A N 011     53
    ! SKS-Wireless                   A O 009   4437
    ! SKS-ZG                         A O 004   4566
    . SentaNET-V-gradska             A N 011   1100
    ! SentaNET5-H-gradska            A N 007   1475
```

Kismet should show a list of available wireless networks.
To sort the press “s” and then select how you want them to be sorted. In this way, you can scroll through them.
Kismet can detect the IP range used inside the network.
Press “i”, and you will see detailed information about the network you selected.
Press “d” to dump packets. These are the basics.
Pressing “h” will pop-up the help window. Everything else is explained here, no need to talk more.

Note that some wireless networks are protected using technologies like WEP,WPA or LEAP. This means you will need some kind of key to verify yourself and get access to the network.
Tools like Asleap (for LEAP) , AirSnort (for WEP) and such will help you do it.

Set MAC address

You might want to do this if the network allows only specific MAC addresses.
One solution is to install a package called macchanger, and another is to edit /etc/network/interfaces.
Open it with a text editor and you should see something like:

```
auto wlan0
iface wlan0 inet dhcp
```

Change it to:

```
auto wlan0
iface wlan0 inet dhcp
   hwaddress ether 01:02:03:04:05:06
```

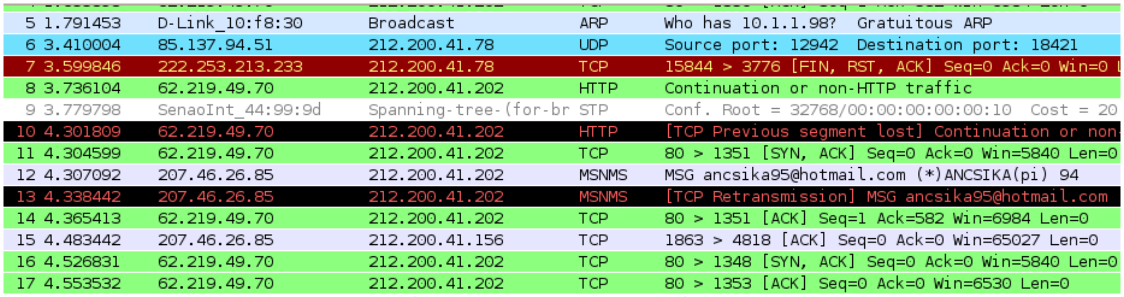
Of source, instead of 01:02:03:04:05:06 write the MAC address you need and instead of wlan0 write the network adapter you will use to connect to the network.
Re-enable your network adapter.

Capturing packets

In order to capture packets , we're going to use WireShark
(previously known as Ethereal)
Let's start:

Get the package from “Add/Remove....” and install it.
Now fire up WireShark as root.
Click Capture -> Options or press Ctrl+K.
In the “Interface” section, write “wlan0” or whatever is the name of your wireless device.
To see a list of your network devices, go to Capture -> Interfaces.
You might want WireShark to write captured stuff to a file.
This is useful if you want to analyze the results somewhere else, or if you're doing long-time captures.
For the best performance, disable the
“Update list of packets in real time” option and don't use complex filters.
Now click Start.

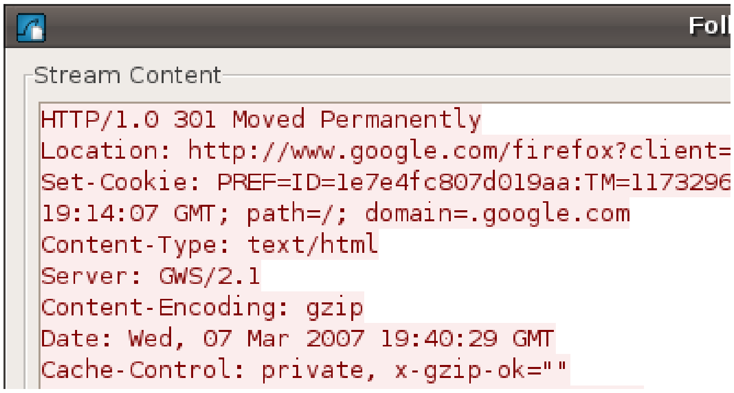
A small window will pop-up. It shows you the amount of traffic that belongs to different protocols in percentage.



```
 5 1.791453   D-Link_10:f8:30     Broadcast            ARP    Who has 10.1.1.98?  Gratuitous ARP
 6 3.410004   85.137.94.51        212.200.41.78        UDP    Source port: 12942  Destination port: 18421
 7 3.599846   222.253.213.233     212.200.41.78        TCP    15844 > 3776 [FIN, RST, ACK] Seq=0 Ack=0 Win=0 L
 8 3.736104   62.219.49.70        212.200.41.202       HTTP   Continuation or non-HTTP traffic
 9 3.779798   SenaoInt_44:99:9d   Spanning-tree-(for-br STP   Conf. Root = 32768/00:00:00:00:00:10  Cost = 20
10 4.301809   62.219.49.70        212.200.41.202       HTTP   [TCP Previous segment lost] Continuation or non-
11 4.304599   62.219.49.70        212.200.41.202       TCP    80 > 1351 [SYN, ACK] Seq=0 Ack=0 Win=5840 Len=0
12 4.307092   207.46.26.85        212.200.41.202       MSNMS  MSG ancsika95@hotmail.com (*)ANCSIKA(pi) 94
13 4.338442   207.46.26.85        212.200.41.202       MSNMS  [TCP Retransmission] MSG ancsika95@hotmail.com
14 4.365413   62.219.49.70        212.200.41.202       TCP    80 > 1351 [ACK] Seq=1 Ack=582 Win=6984 Len=0
15 4.483442   207.46.26.85        212.200.41.156       TCP    1863 > 4818 [ACK] Seq=0 Ack=0 Win=65027 Len=0
16 4.526831   62.219.49.70        212.200.41.202       TCP    80 > 1348 [SYN, ACK] Seq=0 Ack=0 Win=5840 Len=0
17 4.553532   62.219.49.70        212.200.41.202       TCP    80 > 1353 [ACK] Seq=0 Ack=0 Win=6530 Len=0
```

In the main window you can see a list of captured packets, their number, the time that passed since the beginning of the capture, the source IP (the packet is sent from that IP), the destination IP (the IP that receives the packet), protocol, and some info about the packet.

After a while, stop the capture. Let's take a look at the data.



```
HTTP/1.0 301 Moved Permanently
Location: http://www.google.com/firefox?client=
Set-Cookie: PREF=ID=1e7e4fc807d019aa:TM=1173296
19:14:07 GMT; path=/; domain=.google.com
Content-Type: text/html
Server: GWS/2.1
Content-Encoding: gzip
Date: Wed, 07 Mar 2007 19:40:29 GMT
Cache-Control: private, x-gzip-ok=""
```

Right click a packet from the list (HTTP protocol, for example) and select “Follow TCP Stream” to link all packets that are sent between two computers and belong to a single protocol.

It will be easier to read now.

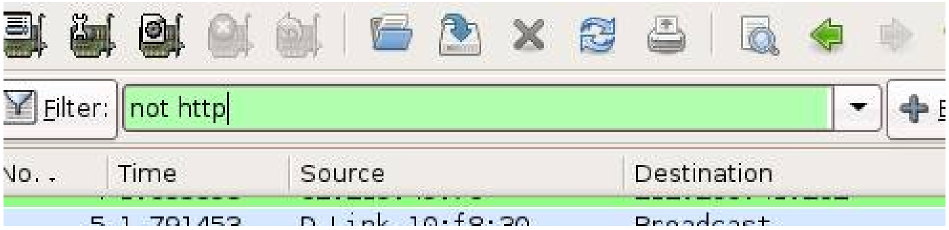
Now you can see fragments of webpages and HTTP replies/requests.

Doing this with protocols that don't encrypt login information (FTP, IRC, POP, ICQ, AIM, Telnet, etc.) you will probably be able to see passwords in plain-text.

Also, by following streams of packets that belong to chat protocols (MSNMS, AIM), you will be able to monitor chat conversations.

Protocols such as HTTPS or SSH are encrypted, so you can't sniff logins with ease.

You might want to sniff packets between two particular hosts only, or between two ports of a specific protocol.



Filter: not http

You can do that by right clicking a packet and selecting Apply as Filter, or by writing specific rules in the Filter textbox.

Some examples (write the without brackets):

“http” - Only packets that are sent through HTTP protocol
“ip.dst == 192.168.0.2” - Only packets sent to 192.168.0.2
“ip.src == 192.168.0.2” - Only packets received by 192.168.0.2
“tcp.port == 80” - Only packets that belong to TCP port 80
“not udp” - Everything except UDP
“aim || irc” - AIM or IRC

You can find more examples and filtering options in Analyze > Display Filters.

All in all, that's the basic usage of WireShark. Read Help > Contents or press F1 for more info.

Conclusion

This is only the beginning ;)

For more information about WLANs and sniffing, here are some useful links:

http://www.nongnu.org/orinoco/ - Orinoco drivers
http://airsnort.shmoo.com/ - AirSnort
http://monkey.org/~dugsong/dsniff/ - dSniff
http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html – Linux WLAN
http://www.goonda.org/wireless/prism2/ - Linux and Prism2 based wireless cards
http://www.kismetwireless.net/ - Kismet
http://www.wireshark.org/ - WireShark
http://en.wikipedia.org/wiki/IEEE_802.11 – Description of the 802.11 standard
http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy – How WEP works

Tutorial by Feky - http://feky.bizhat.com
fekyweb@gmail.com