



Kyle McDonald [Follow](#)

Artist working with code.

2 days ago · 4 min read

How to get free wifi on public networks



Photo Credit

This short tutorial describes a few methods for gaining access to the internet, a basic human right, from public wireless networks.

This tutorial has been tested on Mac, should work on Linux, and hasn't been tested on Windows.

Preparation

Make sure you do this step *before* you are stuck without Internet access. Enter the following commands into your terminal to download and install my open source FreeWifi Python script.

```
git clone https://github.com/kylemcdonald/FreeWifi  
cd FreeWifi && pip install -r requirements.txt
```

How to get additional time

If you had free internet access but your time has run out, the first thing to try is open an incognito/private window. Here are instructions for a few browsers:

- Chrome (mobile and desktop)
- Safari for iOS
- Safari for Mac
- Microsoft Edge

An incognito/private window will temporarily clear any cookies that may have been used for tracking how much time you spent online, making you look like a “new user” and allowing you to log into the wireless portal again.

Unfortunately, most systems track MAC addresses instead of cookies. A MAC address is a unique identifier assigned to every network interface. This means you need to get a new MAC address to get additional time. Fortunately, MAC addresses can be changed in software, without swapping the hardware. The `sudo spoof-mac randomize WiFi` command line utility makes this easy by entering `sudo spoof-mac randomize WiFi`. If the command fails to run, try entering `spoof-mac list --wifi` to check what the name of your wireless device is first, and use that manually. After randomizing your MAC, try logging into the wireless

portal again. When you're done using the Internet, run `sudo spoof-mac reset Wi-Fi` to reset your MAC address.

Note that MAC address spoofing may be interpreted as an illegal activity depending on why you do it. In some cases it is certainly not illegal: recent mobile operating systems like iOS 8+ and Android 6+ automatically randomize their MAC address when searching for wireless networks to avoid being tracked. But when [Aaron Swartz liberated JSTOR](#), MAC address spoofing was claimed as a signal of intention to commit a crime.

How to get free access

If the network is open, but you can't get access for some reason, you can also try spoofing the MAC address of a device that is already using the network. To the router, your device and the other device will look like one device. This can cause some minor problems if they interrupt each other, but for light browsing it usually works out fine.

To find the MAC addresses of other devices using the network, first you need to connect to the network. You don't need to have Internet access, just a connection. Run the command `sudo chmod o+r /dev/bpf*` once to make sure you can sniff wireless data (you only need to do this again if you restart your computer).

Then in your terminal, run the command `python wifi-users.py`. You should see a progress bar immediately:

```
SSID: nonoinflight
Gateway: 00:e0:4b:22:96:d9
100%|██████████| 1000/1000 [00:46<00:00,
21.46it/s]
Total of 5 user(s):
27:35:96:a8:66:7f    6359 bytes
36:fe:83:9c:35:eb    9605 bytes
65:01:3c:cc:20:e8    17306 bytes
8c:6f:11:2c:f0:ee   20515 bytes
0a:4f:b2:b8:e8:56   71541 bytes
```

If there isn't much traffic on the network, it might take longer. If it's taking too long, type `CTRL-C` to cancel the sniffing and print whatever results are available. Finally, we want to spoof one of these

MAC addresses. For example, in this case we would enter `sudo spoof-mac set 0a:4f:b2:b8:e8:56 Wi-Fi` to try spoofing the address with the most traffic (they probably have a connection).

After running this command, try to access the Internet. If you don't have a connection, try the next MAC in the list. If your Internet connection drops out while using this MAC address, try disconnecting and reconnecting to the wireless network. Note that the original user of the MAC you copied may experience these same connection drop outs if you are both actively using the network.

How it works

`wifi-users.py` uses `tcpdump` to collect wireless packets. Then we look through these packets for any hints of the MAC address (BSSID) of our wireless network. Finally, we look for data packets that mention a user's MAC as well as the network BSSID (or the network gateway), and take note of that MAC using some amount of data. Then we sort the user's MACs by the total amount of data and print them out.

Instead of sniffing wireless traffic, in some situations you can also use the command `arp -a` to get a list of MAC addresses of devices on the wireless network. Then you can either use `spoof-mac` to copy the address, or use `ifconfig` directly on Linux and OSX. For the specifics of using `ifconfig` look at the implementations of `set_interface_mac` inside `SpoofMac's interfaces.py`.

This post is dedicated to Lauren McCarthy, who has taught me the most about the art of getting a good deal.

