Build a Pi-powered drone disabler to understand the security risks of wireless communications

- By Brent Chapman
- Difficulty: Moderate

2k 151 139 92 9



Photo by Mike Senese

Please note: *The information presented here is for educational purposes. As with all guides covering network and computer security, the techniques should only be performed on devices that you own or have permission to operate on. This tutorial is designed to help users understand the security implications of using unprotected wireless communications by exploring its use in a popular drone model: the Parrot AR.Drone 2.0.*

It's illegal to access computer systems that you don't own or to damage other people's property. As we continue the public dialogue on drone regulations, it's critical to understand as many aspects of the issue as we can to include social impact, policy, privacy and of course, security. We hope that manufacturers take steps to improve the security of their products and users continue to educate themselves on the capabilities and vulnerabilities of emerging technologies. Make: and the author take no responsibility resulting from the inappropriate or illegal actions that result from abuse of any of the techniques discussed.

==========

Quadcopters capable of transmitting high-quality video are making it possible to affordably record unique perspectives. But these "unmanned aircraft systems," as the FAA calls them, have posed new challenges in security, safety, and privacy, and many experts caution pilots to consider the implications of increased drone usage. In addition to the concern of constant surveillance, there's the possibility that businesses (or hackers) can collect location information from mobile devices by using roving drones.

As a result, a cottage industry is forming for anti-drone technology. These devices come in a range of sizes, from plane-mounted to handheld tools. I will show you how to build our own rig to execute a particular network-based attack against one type of quadcopter control: Wi-Fi.

A Word of Caution

While I won't touch on signal jamming or directed energy, it's worth noting that jamming creates serious safety risks and is illegal. Additionally, the computer-based techniques that we'll cover should only be done on networks and devices that you own, or have permission to experiment on.

Why 802.11?

Wi-Fi is a key interface for many current quadcopters. Some use it as the interface between the controller and a tablet displaying mapping and telemetry data. A few drones, such as Parrot's Bebop and AR.Drone 2.0, are entirely controlled via Wi-Fi. This type of system lowers the barriers to entry into the drone space since pilots can use their own devices for control, but it does create interesting security situations since existing network-based attacks can now be used against these devices. Modern drones are essentially flying computers, so many of the attacks that were developed for use against traditional computer systems are also effective. The AR.Drone 2.0 in particular has many impressive features and sensors that users can access, and its low cost makes it an ideal platform for experimentation and learning.

How it Works

The AR.Drone 2.0 creates an access point that the user can connect to via a smartphone. The access point that it creates is named ardrone2_ followed by a random number. This access point by default is open and offers no authentication or encryption. Once a user connects the device to the access point, he or she can launch the app to begin control of the drone. This process, though convenient for the user, makes it easy to take control of the drone. The AR.Drone 2.0 is so hackable, in fact, that there are communities and competitions focused on modifying this particular drone.

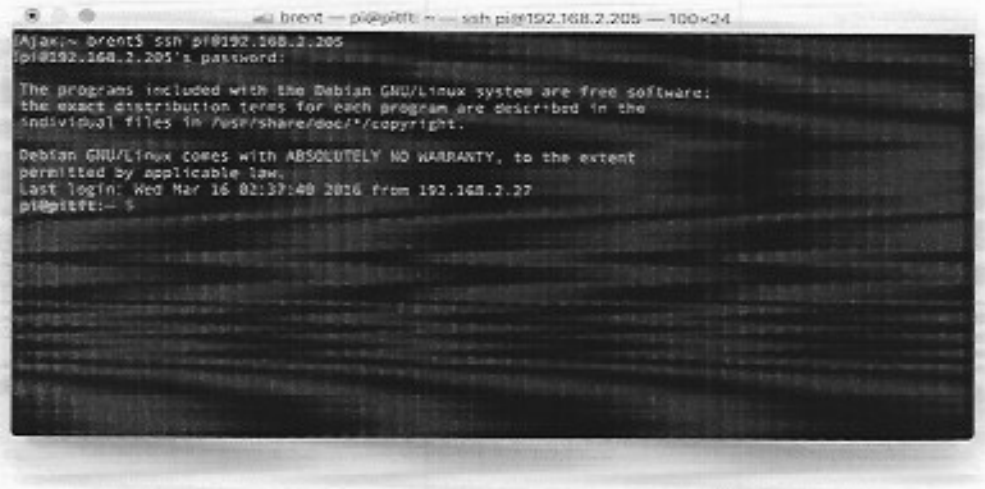
Our Test

Using a laptop computer, USB Wi-Fi card, and our new antenna, we'll explore a very simple attack. Power on the AR.Drone 2.0 and have a friend fly it around using the app. After a few seconds, its access point should also show up in your available wireless networks. Connect to the network and start up your favorite terminal application. The default gateway address for this network will have an address of 192.168.1.1. You'll be able to telnet to this address since the service is, unfortunately, left wide open on this system.
Telnet is an older protocol for accessing remote computers. At this point, you can explore the system, or shut it off entirely without the legitimate user knowing what's going on. Using a combination of freely available network tools, you can easily perform all these steps from your computer.

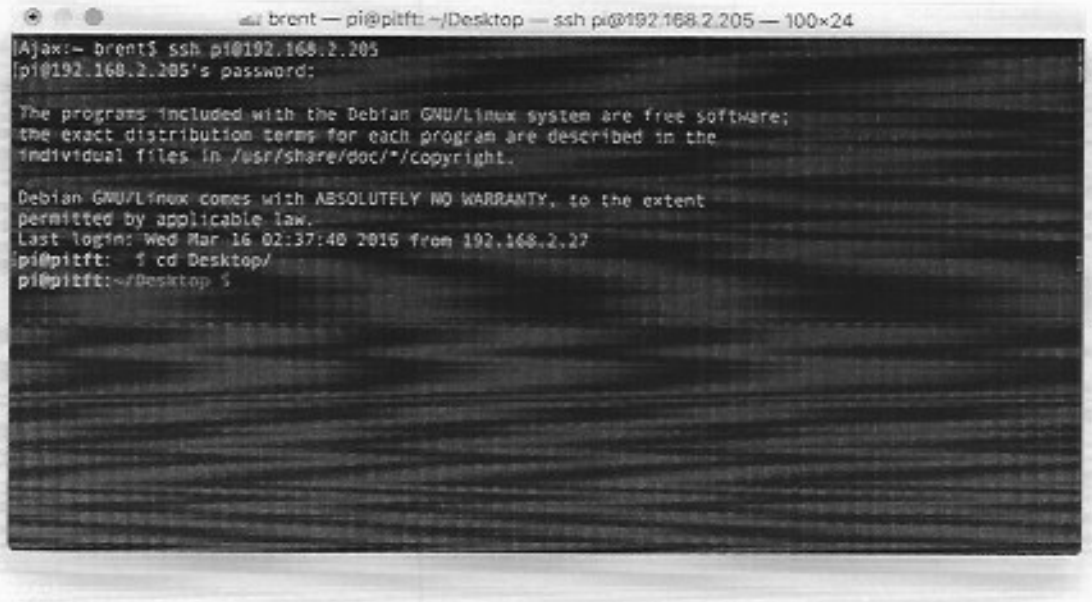
Now we'll look at how you might automate this attack with a Raspberry Pi, a touchscreen, and a couple of Bash scripts.

I used a great tutorial provided by Adafruit (learn.adafruit.com/adafruit-pitft-28-inch-resistive-touchscreen-display-raspberry-pi) to set up my Raspberry Pi with a touchscreen, so that I could launch my attacks with a click. Assuming that you have a Pi already set up, let's walk through how you could automate this.

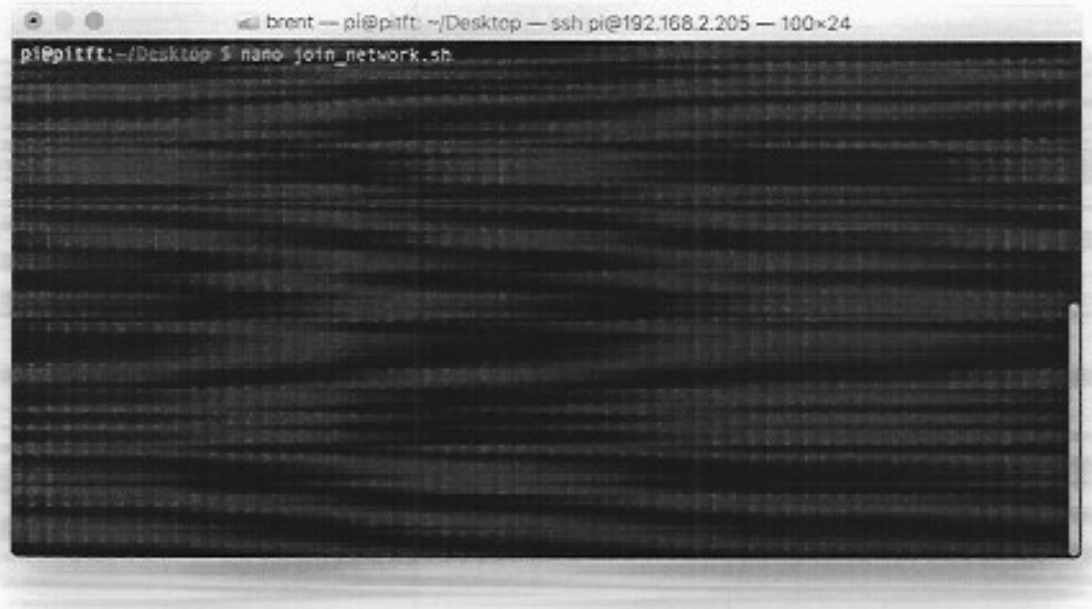
The first step is to log into your Pi using SSH.



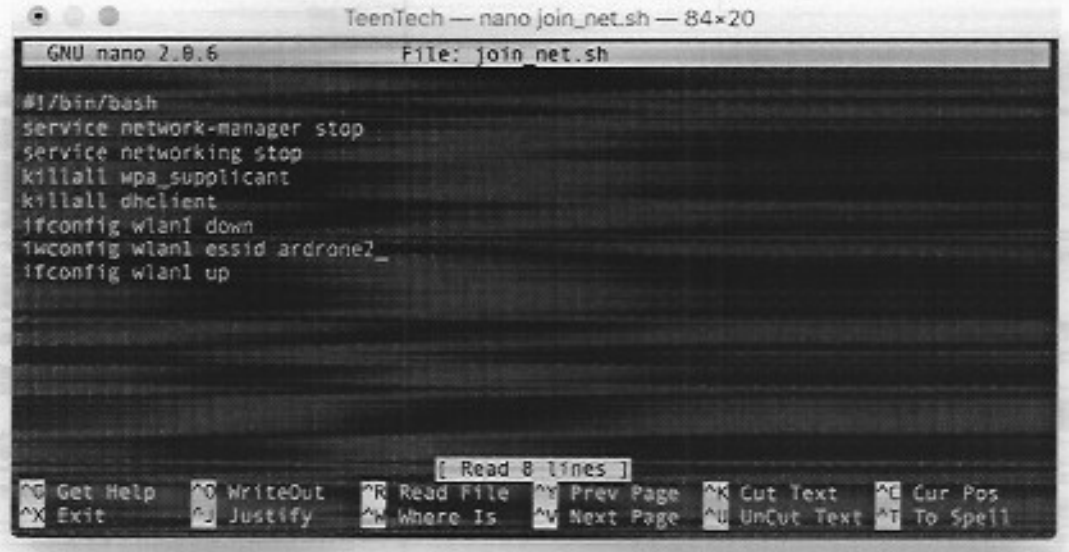
Change directory to the Pi's desktop (or wherever you want) so that the scripts are easy to find and click.



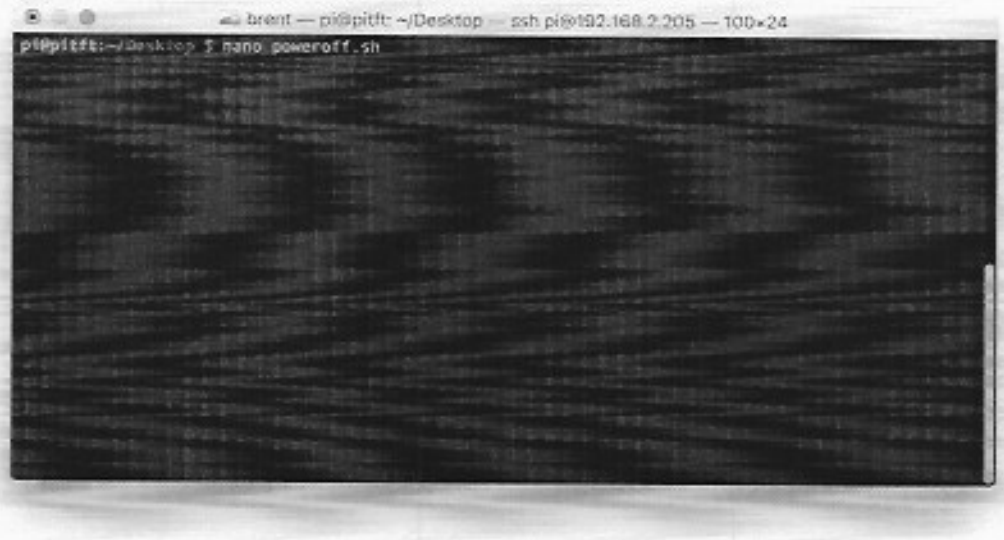
Using your favorite text editor, create a new file. I named this join_network.sh because I'll be using this to make the Pi automatically join the AR.Drone 2.0 access point.



Add these 8 lines to your script. On line 7, enter the full name of the AR.Drone 2.0 access point. Once you're done, save everything.



You're now going to automate the connection that you tested before and send an additional command to shut the drone down. Start by creating another script. I called mine poweroff.sh.



Add these lines to your script. This initiates a telnet connection to the drone, which is located at 192.168.1.1, and sends the command of poweroff, which tells the drone (which is a computer after

Now make sure that the scripts are executable. Do this by typing sudo chmod u+x filename. Check this for both of the files; we can verify that they are now executable by typing ls -la and looking for

The two scripts are ready to use. Be sure that no people or fragile items are below the drone when you're testing. Have fun!

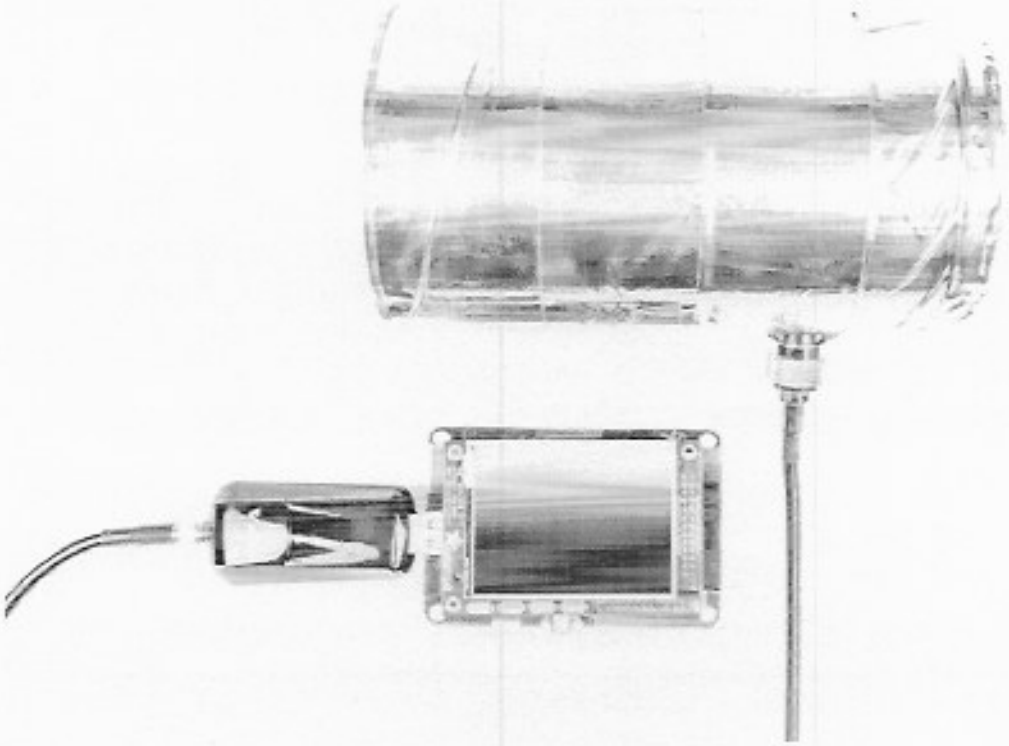


Photo by Hep Svadja

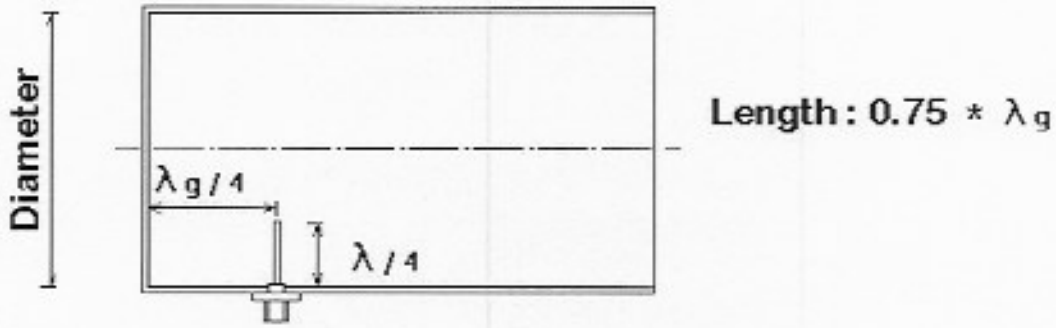
Build a Cantenna

Boost your wireless signal with a directional antenna made from a can

In a wireless world, connectivity is king. A good antenna attached to your wireless device will boost your signal and dramatically extend your range. In less than an hour, you can build your own directional "cantenna" to connect to distant wireless hot spots or interact with wireless devices like some of the drones featured in this issue.

1. Calculate

The toughest part about this build is calculating the best location for mounting the radio connector, and the correct length of the wire element for ideal performance of the antenna. Fortunately, there are lots of online resources to help you with the math, such as csgnetwork.com/antennawncalc.html. The illustration below gives an overview of how the measurements are calculated.



Given the dimensions of the can, about 100mm in diameter, the Type N connector needs to be mounted 44mm from the bottom of the can. The frequency we're interested in is in the 2.4GHz band, so the total height of the copper wire needs be roughly 31mm.

2. Measure and mark



Step shots by Brent Chapman

Measure 44mm up from the bottom of your can, and mark the position for the N connector with a permanent marker. (I measured 44mm down from the top of my cookie tin, which has a replaceable lid that I used as the back of my antenna.)

3. Drill holes

On the mark you made, drill a hole so that your N connector can fit snugly. It's good to start with a small bit and work your way up until the hole is just large enough. Once you're done, sand the area around the hole to ensure good contact with the connector.

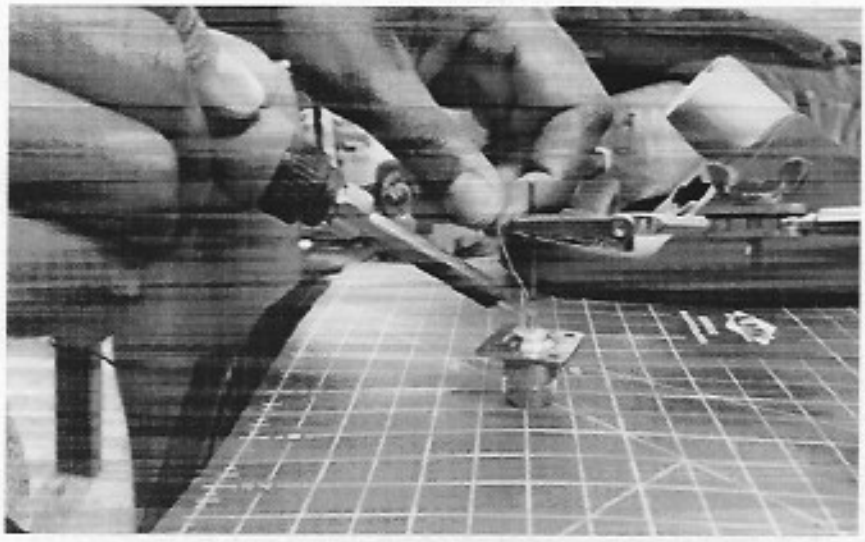


Test-fit the connector and mark the 4 mounting holes. Drill these to match the machine screws you'll use to mount the connector. Or skip the screws and just solder the connector to the can.

4. Solder the wire to the N connector

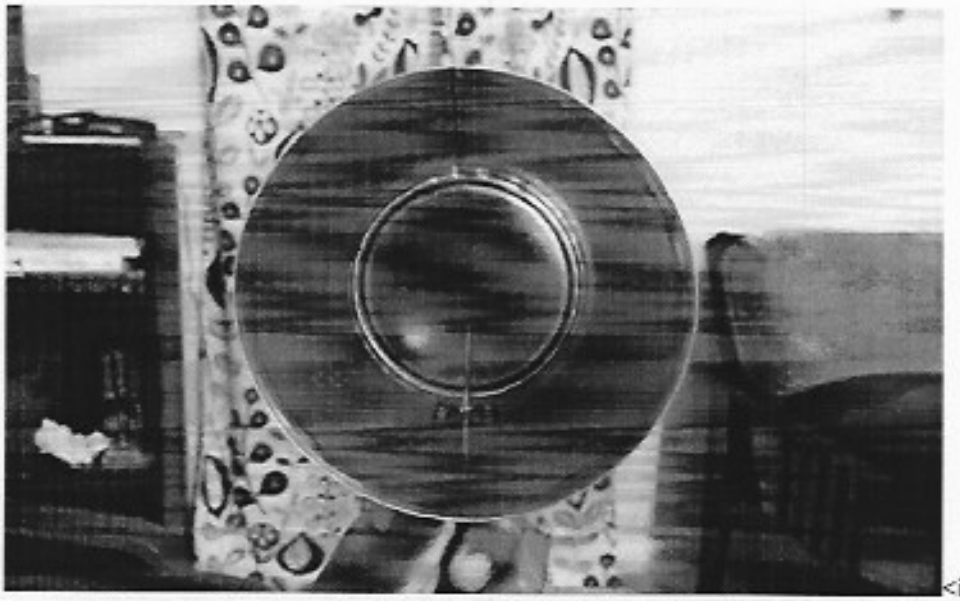
You need to prepare the connector before it's mounted. Take a 4" piece of straight copper wire — the straighter the better — and remove any coatings.

Now you're going to solder that short copper wire to the top of the connector. It's a little tricky; I used helping hands to position everything before soldering it in place.



After soldering the wire to the connector, test-fit again and then trim the wire to the distance you calculated in Step 1. In my case, that was 31mm.

5. Mount the Connector



If you didn't solder the connector to the can, tighten the machine screws from the outside of the can into their nuts inside. If needed, you can cut the bottom off the can for access, then tape it back in place when you're done. There — you have a brand new cantenna.

6. Connect to Wi-Fi card and enjoy

Screw the pigtail cable into your card and the N connector. Your cantenna is ready to use.

Going Further

You can add a coat of paint to make it more tactical, or add a handle or mount it on a tripod for precise aiming.

Other Drone-Related Possibilities

This is just the tip of the iceberg — there are a number of things that an attacker could do. These include modifying or deleting system files, intercepting video and sensor feeds, rerouting the drone to alternate locations, or a combination of these. Hacker and Maker Samy Kamkar, the person behind security projects like RollJam and MagSpoof, even released a project designed to allow an attacker drone to autonomously seek out any Parrot drones within Wi-Fi range, disconnect the real user and initiate a new connection that is controlled by the attacker drone. The end result essentially is an army of "zombie" drones.

We also tested a range of drones at the Make: office that rely on some form of Wi-Fi connectivity for their operation. All of the drones tested were susceptible to deauthentication (deauth) and disassociation attacks, which forced all users off the drone's access point, resulting in a loss of connectivity to the drone.

Beyond Drones

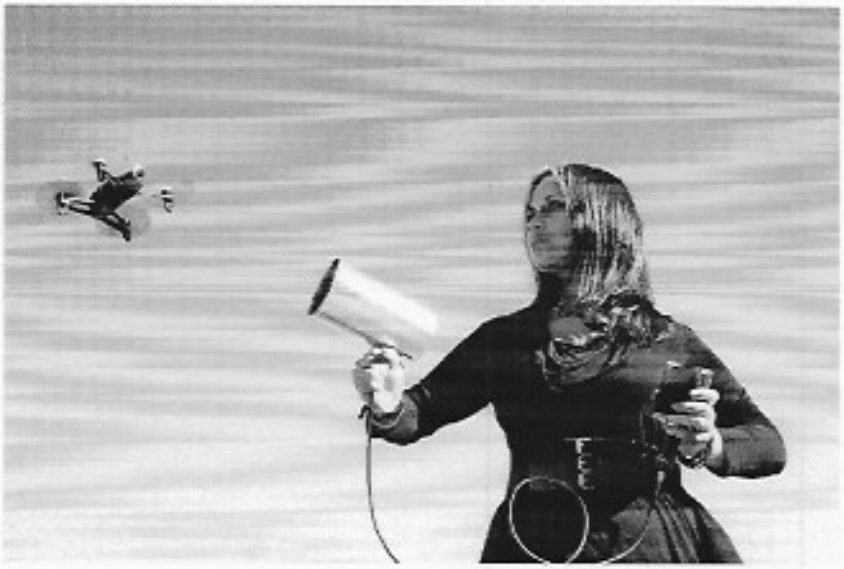


Photo by Hep Svadja

The DIY "cantenna" is incredibly useful for vastly extending range of connectivity. Using the Raspberry Pi rig we've just assembled, an attacker could reprogram the computer to perform a number of attacks, such as a deauth attack against a coffee shop hot spot. How is this useful? Well, consider the following scenario: An attacker sets up a fake access point called "Better Wi-Fi" that is designed to collect credentials. Customers are content using the real coffee shop's connection so there's no reason for them to join the attacker's fake network. Knowing this, the attacker uses his rig to deliver the deauth against the real access point to force all the users off. The users can no longer reach the real access point, and in need of internet connectivity they connect to the evil (but convincing sounding) hot spot and their account credentials are collected.

How to Protect Yourself

The first step, of course, is educating yourself on the capabilities of your drone, its limitations, and good security practices. There are advantages to using Wi-Fi, for example, as the means to control machines, but there are many things to consider from a security point of view, such as wireless security protocols, encryption, and open ports. For more sensitive applications, there are far more secure options when it comes to command-and-control. Always ask permission and tinker safely!