About this document:

This document looks a lot longer than it actually is, there's lots of whitespace, I'm excessively verbose, and a lot of it is explaining what terms mean, *why* we're doing what we're doing, and some sprinklings of esoteric knowledge gleaned from 20+ years of fighting with tech. It's a lot of little things that you may not have done before that just add up. Once you've done it this first time though, most of your time on subsequent devices will be in finding the firmware and then waiting while it transfers and installs.

What you'll need:

- A computer with some way to connect to a wired network (built in or USB NIC)
- A Power over Ethernet injector or switch. (We have some you can use during con)
- (Optional) A serial console cable. (Again, we have some you can use during con)

A note about CTF flags:

There are three CTF flags that you can find while going through this guide. I do recommend submitting them as not only will they score you some points they also serve as a way you can check your work as you go along. To submit the flags you will need to replace the text between the {} with what you've found.

Step 0- Figure out what you have.

Look for labels with manufacturer names, model numbers, serial numbers, FCC IDs, MAC addresses, any little thread you can start pulling to get more information. In this case we've got "Cisco" in a couple places and an obvious label telling us "Model: AIR-CAP2602I-A-K9"

Step 1- Find info.

Use your favorite search engine, head to the manufacturer's website, generally just gather as much documentation as you can. In some cases, you'll be able to just download the firmware and step-by-step instructions on how to flash it straight from the manufacturer. Often though, you'll find enterprise networking vendors are jerks who hide stuff behind account logins and “Service Contracts”. Basic accounts are usually free and open up a bit more info, if not access to downloads; but, in this case, we aren’t even going to need that.

Fortunately, with Cisco, a bit of poking around (even without an account) in the “Downloads” section will get you what we need; the filenames and, more importantly, the MD5 and SHA512 hashes of their various firmware files… even if they won’t let you download them.

But what’s this “Autonomous”, “IOS Boot Images”, “Lightweight” stuff? For our purposes; “Autonomous” is for standalone APs (what we want); “Boot Images” are for trying to de-brick things; “Lightweight” is for installations that use a Wireless Lan Controller.

CTF Flag: pdx{the SHA512 of the latest Autonomous firmware}.

Step 2- Find and verify firmware.

Note: even when they don’t make it publicly available, it is often possible to contact the manufacturer’s support and social engineer them into giving you the firmware you need… but it usually takes more than two days, so we did it for you this time.

DISCLAIMER: obtaining software / firmware through “alternate means” from those approved by the manufacturer is a legally grey area. Support Right to Repair.

I pointed out the importance of the file hashes because those allow us to get our firmware files from the *sketchiest* of sources and still be confident it’s not backdoored or otherwise tampered with. Like, say, from “a random USB drive you were given at a hacker con” or “a sketchy filehosting service based in a country with no regard for US copyright law that you found linked to by a random forum post after you dropped the filenames and/or hashes into your favorite search engine”.

Speaking of… the drive you got with your AP contains real firmwares obtained in an entirely legal manner… but it also has some bogus versions! Don’t worry, these are all benign (either older firmware [which you may want to keep] or just full of nuls); stuff downloaded from the internet might not be. To figure out which are the good firmwares you’re going to use those hashes.

What the heck is a hash? For our purposes, they’re a whole bunch of math that will tell you if two inputs are *absolutely identical* (e.g. if the file you have’s hash matches the one Cisco published, it means your file is a bit-perfect copy of the file they have; and that means nobody has tampered with it). Fortunately, we can make the computer do all that math for us.

There are lots of programs you can download to take a file as input and generate a hash of it. But, you probably already have all the software you need. On Windows you have the Get-FileHash module in PowerShell, md5 and shasum in BSD terminals (including MacOS), or md5sum and sha512sum in most Linux distributions. Compare the outputs from those to the MD5 and SHA512s from Cisco; if they don't match, that's one of the bogus firmwares!

CTF Flag: pdx{the SHA256 of the latest Autonomous firmware}.

Step 3- Connect to the console (optional but *highly* recommended)

This isn't strictly necessary for flashing the firmware, but this is how you can see what's on there now, watch stuff happen, and interact with the device's CLI. It's good stuff to know because this is how you get back into a switch after you accidentally turn SSH off… not that any of us have ever done that…

Step 4- POWERRRR!!

Connect the ETHERNET port of the AP to a Power Over Ethernet source (either a PoE enabled port on a switch or the "out" port of an injector).

If you've got a console connection open, you'll see a bunch of stuff show up. There are often interesting bits of information in this output (even if you don't have credentials); things like the version of the firmware the device is currently running, some details about the network environment it's expecting, etc. (Side Note: this is why it's important to properly decommission devices by wiping them back to factory settings when you're disposing of equipment… a lot of places don't actually do that though… ebay accordingly).

CTF Flag: pdx{The flash image name for currently installed firmware}.

If you don't have a console connection open, this just serves as an "is it dead" test. The LED on the front should light up and blink various colors.

Step 4b: Boot into recovery mode (optional for now)

Hold down the "MODE" button while plugging in power, after ~20-30 seconds the LED on the front will turn red and there will be a "Button is Pressed" message in the console, at that point you can let go of the button.

Step 5- Pick a TFTP server.

I recommend Tftpd64 for Windows, it's FOSS and works better than Microsoft's that you have to install anyway. MacOS has a built-in terminal tool, and there are several options on Linux. I put this here because we're going to be messing with network interfaces in the next step and internet access *can* get a little wonky when we do that, so grab a local copy of any instructions you need.

Step 6- Configure wired networking.

When you boot the device into recovery mode it's going to default to a specific IP in one of the local IPv4 blocks (10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16) and then go looking for a tftp server in that range. The easiest way to make stuff work is to just statically assign an IP in that range to a wired NIC. A lot of modern laptops no longer have RJ45 jacks, so you may need a USB NIC.

The specific IP can vary, but 10.0.0.1 is the most common in my experience. Setting your NIC to 10.0.0.2/8 with a 10.0.0.1 gateway is a good place to start (or you could do something crazy like read documentation).

Note: if you did Step 4b, the console output tells you not just the IP the AP uses, but the filename it's going to look for on the TFTP (next step).

Step 7- Configure the TFTP server.

You need to configure your TFTP server to listen on the interface you set up in Step 6 and host a copy of your firmware file which has been renamed to whatever the device is looking for (this can be found from documentation or through console output). In this case we need to rename it to ap3g2-k9w7-tar.default (Note: this is the same regardless of whether you're flashing Lightweight or Autonomous firmware).

Remember to allow access through any host-based firewalls.

Step 8- It's Go Time!

With your TFTP server running, your NIC connected to the "In"/"Switch" side of a PoE injector or to a PoE switch, and (optionally) a console session open; it's time to finally flash this thing! Go do Step 4b, it's no longer optional.

Note: If you are doing this during con, we have a few PoE injectors next to our volunteers if you want to catch a talk during the next step without leaving your stuff *completely* unattended at a hacker con.

Step 9- Wait.

Seriously, it's going to take around 20 minutes. Depending on your TFTP server, you might see some status or upload progress information. If you're watching the console, you'll see messages. The LED will blink green while it's doing its thing and then be a nice steady green once everything has installed and the device has rebooted.

Troubleshooting: you may need to increase the timeout and retry settings of your TFTP server; these APs sometimes take a bit longer to respond than the server programs expect by default. If it times out, the process will fail and the LED will blink red… bump up the timeout and try again.