DAVID SHAYER 4 August 2020 No comments

Understanding How Apple Security Research Devices Likely Work and Stay Secure

Apple recently announced the Security Research Device Program (SRD), in which select security researchers receive special iPhones to help them root out iOS security problems (see "Apple Releases Dedicated Security Research Device," 23 July 2020). Apple can then fix any security vulnerabilities they find, hopefully before they're exploited.

Based on Apple's announcement, backed up by some logical deduction, we can speculate about how this program and the particular devices will work.

When you buy an iPhone from Apple, it's a production phone, with a production code signing key burned into its system on a chip (SoC). Having a production key is referred to as *production fused* (often shortened to *prod fused*). It runs release versions of iOS, which have been signed with Apple's *production certificate* (internally called the *prod cert*). When the iPhone boots, code burned into the SoC checks the operating system. If it's not properly signed with a prod cert, the iPhone refuses to boot at all. Jailbreaking is (partly) the art of figuring a way around this restriction.

Similarly, iOS will run only apps that have been signed by the production App Store cert, which prevents sideloading, the option of installing apps directly, without going through the App Store. There are a few exceptions, such as a third-party developer signing an app they build with their third-party development cert, which allows the app to run on their iPhones. In practice, this process is often fraught with problems. (This description greatly simplifies the details of code signing, but it covers the general procedure.)

Developing software internally at Apple would be extremely difficult with these restrictions in place. Internally, Apple engineers use iPhones with a development key (they're referred to as *dev fused*), and they run development builds of iOS. Dev builds of iOS include a shell, debugging and profiling code inside many apps, test hooks, and internal frameworks. Dev iOS builds are signed with a *dev cert*, so they boot on dev-fused iPhones.

Additionally, dev builds of iOS don't check an app's cert before running it. You can load any app you want onto a dev-fused iPhone. This fact makes it a lot easier for Apple engineers to do their work. One reason the process for signing apps works poorly for third-party developers is probably because Apple engineers don't use it, so they don't apply any pressure to fix problems.

Even if you managed to get a copy of an iOS development build and somehow load it onto your production iPhone, it wouldn't boot, because it's not signed with a production cert. Likewise, release builds of iOS won't boot on dev-fused iPhones.

As I noted, dev builds of iOS include a shell, the app behind Terminal that runs Unix commands and scripts. macOS and iOS used to default to the bash shell. macOS recently moved to zsh; iOS may have moved to zsh too.

Since an iPhone is a full-blown computer, Apple engineers log into their dev iPhones using ssh and work in the shell. It's no different than server engineers ssh'ing into a remote server to work on server code. It's very difficult to work on a computer if you can't log in to it. Release versions of iOS don't include a shell, specifically to improve security (and make life difficult for jailbreakers), since there's nothing to log into.

If Apple wants to help security researchers find vulnerabilities, Apple must give them iPhones with a shell they can ssh into. I've heard unofficially that the SRD iPhones are neither production nor internal development, but are probably iPhone 11s with a new SRD cert—call them *SRD fused*. They'll run a special iOS build that has some of the features in Apple's internal iOS builds and is signed with the SRD cert. This implies that the SRD build of iOS won't run on production iPhones, nor on Apple internal dev iPhones.

Apple is also likely providing researchers with a special internal build of Xcode, plus tools designed for loading software onto iPhones and debugging the internals of iOS. Apple has a wide range of internal tools for spelunking into the dark recesses of iOS where normal third-party developers never go. These tools would definitely help security researchers. Apple makes new internal builds of iOS, Xcode, and support frameworks daily. While security researchers are unlikely to get daily builds, they'll probably get regular updates.

SRD builds of iOS probably also run apps without checking that they're code-signed, just like iOS dev builds. I don't have any specific information indicating this is true, but it would make sense. This would allow a security researcher to set any entitlements they want on an app and learn a lot about how iOS's internal protections actually work. It would also mean they don't have to fight with Xcode's code signing feature every time they load an app onto the iPhone.

iOS has many levels of security. A fully working exploit is usually built by chaining together several individual vulnerabilities. For instance, one vulnerability may allow a malformed JPEG to take over the JPEG decoder. But the JPEG decoder is sandboxed, specifically so that if it's compromised, it can't access very much. Another vulnerability might allow escaping the sandbox. A third vulnerability might allow escalation to root. Chain these three together, and you have an exploit, where just looking at a picture compromises your iPhone. (That's called a "drive-by exploit.")

While a few of the dozens of iOS security restrictions will be turned off in the SRD builds of iOS so security researchers can more easily do their work, most will remain in place. Apple doesn't need security researchers to find full-blown, multi-stage exploits. Apple will be happy if the researchers find ways around individual security restrictions so its engineers can fix those mistakes discreetly.

Have you wondered what prevents bad guys from getting their hands on one of these SRDs? Apple likely has some strong measures to keep the SRDs and special iOS builds from leaking beyond the intended recipients. For starters, there's probably a non-disclosure agreement as tight as anything in the industry. In the past, Apple security has required that people who have access to unreleased hardware must keep it in a locked, windowless room and give keys only to specific people. Apple sometimes uses different code names with different groups, so if a code name leaks, it can be traced.

Inside Apple, unreleased iPhones "phone home" via the Internet every few days, so Apple knows the employee it's assigned to still has it. The SRDs may do this too.

Even though the SRD is likely to be a standard iPhone 11, albeit with a special cert, Apple still may not want pictures of it or screenshots from it circulating. In the past, Apple security has put "random" markings on the cases of unreleased products, so if a picture were to appear on the Internet, Apple would know which device it was. Each SRD build of iOS could have identifying markers added, so if a build were to leak, Apple would know where it came from. An Apple engineer once told me that some internal OS builds use steganography to hide the iPhone's IP address and MAC address in the low order bits of screenshots. The data isn't visible to the naked eye, but if a screenshot appears online, Apple has tools that can see where it came from.

If an SRD were stolen, depending on how the SRD builds of iOS are signed, Apple might be able to revoke the certificate for that device to prevent future iOS builds from running on it. Apple could also likely brick an SRD remotely, though I assume anyone smart enough to steal an SRD would keep it in a Faraday bag and make sure it never appeared on the Internet to receive the brick command.

Recent high-profile iPhone hacks may have prompted Apple to re-evaluate how it does security research. Given how secretive Apple usually is, the Security Research Device Program is an unusual step. As you can see, it's also one that undoubtedly required a significant amount of work to ensure that it couldn't be exploited by organized crime and government intelligence agencies. Hopefully, we'll all get more secure iPhones as a result.

Apple SRD | Security

COMMENTS ABOUT

Understanding How Apple Security Research Devices Likely Work and Stay Secure

Start the discussion in the TidBITS Discourse forum