Post-Evasi0n 7 Breakdown

On December 22nd, the iOS 7 jailbreak named “evasi0n 7” was released. There has been a lot of drama and confusion surrounding this release. Now that the dust has settled, it is time to look back and reflect on what has transpired and how it affects us.

##1. What happened?

The official story from the evad3rs seems to be that they made a partnership with a Chinese company (TaiG) and was to bundle their alternative app store with the release of the jailbreak to (specifically) Chinese users. This was to only be offered to those installing the jailbreak with their computers set to Chinese language/locale. The agreement between evad3rs and TaiG was that the alternative storefront they provided would be a legitimate business and not promote piracy. Upon release of the jailbreak software, it was quickly discovered that evasi0n 7 would install this alternative store for Chinese users, but not anyone else. This sparked a lot of controversy due to the lack of public statement about this as part of the jailbreak. Almost immediately a lot of very unpleasant information surfaces, mainly focusing on evad3rs being involved in financial arrangements in the form of pay-offs and suspicious business arrangements with TaiG, but also betrayal and information leaks within the community of iOS security researchers. Over the past few days we have received two official statements from evad3rs that show some light on the events that took place, however some of that is still in question, especially with testimony from prominent members of the community speaking out against the evad3rs on their actions and explanations. This fiasco took off when it was discovered that the alternative app store provided by TaiG was not only distributing pirated software from both the official App Store, but also pirated software and tweaks that are sold through the Cydia storefront. At first this was claimed to be a mistake, and TaiG was standing with the evad3rs in a move against piracy. However it wasn’t even required to scratch under the surface of their app to see ties to Installous, AppSync, and a number of pro-piracy repos. After that evad3rs made a statement that they were disabling the ability for the TaiG store to be installed via their jailbreak. The following day TaiG makes an unexpected move of releasing an unofficial 1.0.1 patch to the official evasi0n 7 app for windows to bypass checks and force the install of their own store regardless of Chinese language settings. For a while it seemed that TaiG started to block access to their website, where they were distributing this unofficial patch to countries without a majority of Chinese speaking populations. Namely countries that still had access where: China, Taiwan, and Indonesia. As of this post, they still appear to be distributing their patched version of the evasi0n 7 jailbreak. As of last night, evad3rs have released an official 1.0.1 patch which removes the ability to install the TaiG app store and have removed the TaiG app payload from the app. They have posted a response to the community about their actions, and what has transpired.

##2. My Reaction

For those of you that didn’t know of me before this, I have never been a supporter of the JB community. To this day I don’t understand the desire to zero-day my phone. That said, I do understand and respect the need to push the boundaries and to break out of the box that Apple gives us. I woke up on the 22nd to the very beginning of this entire fiasco and following the story as it was being told by Stefan Esser. At the time it appeared to many as though it was just the ugly head of the drama beast being reared. Over the next few hours it was pretty clear that wasn’t the case and the situation was as ugly as it looked. I decided to apply my skills at reverse engineering to the newly released evasi0n 7 app to see if some of the claims about the TaiG store were true. I ended up being one of a handful of people that publicly reversed and exposed parts of the jailbreak.

Reverse engineering is my bread and butter, I love taking apart software to see what makes it go. My work on reversing and patching evasi0n 7 was in the strictest sense of the phrase “fire-fighting”. My goal from the beginning was to help the community feel safe with the software they were using. I patched out the TaiG payloads from the official release and explained how to patch the binary to prevent the installation of TaiG for both Mac and Windows. My work and role in this was been trivial at best. I have been very vocal with this information because of the impact that this would have made beyond the JB community. I was quick to find and point out that the bundled TaiG store was codesigned, typically unnecessary for jailbroken devices. The more significant fact being who the CA was, Apple Inc.. They were using an official iPhone Distribution certificate to codesign the software they were installing on jailbroken devices. Cursory investigation of both the TaiG app binary and helper dylib screamed of Installous and AppSync. I immediately made that fact public, and have taken steps to speak with people at Apple Inc. so that the cert and profiles in the TaiG app were revoked and investigated. I considered this to be mandatory, as it was completely in line with my original intentions. Over the last few days I have posted a couple of binary patches to the original evasi0n 7 mac binary, code for how those patches work, and the changes that can be made to the releases by TaiG and evad3rs to bypass the installation of the TaiG payload.

The one thing that I don’t think I can move on from is the betrayal and disbelief surrounding this whole scenario. Much of it rests on the word of the evad3rs and other community members. The people that have claimed to be against piracy from the start allowed payloads that are specifically for pirating software to be packaged and shipped in an official jailbreak release. It is possible that they screwed up and missed something, many of us aren’t privy to what has happened behind the scenes to say one way or another. I find it hard to accept that they missed that, whatever the reason. I also find it hard to accept that they thought that by simply using tools like o-llvm that others wouldn’t be able to repurpose their release for their own benefit.

As an aside, I would like to extend an apology to those that are responsible for the o-llvm project. You got the ugly side of this and I don’t think you have received an adequate apology for being associated with this mess.

##3. The Jailbreak

The app itself is pretty simple. It has a number of payloads bundled as .tar.gz files. Both the mac and windows versions of evasi0n include these payloads in the binary itself. The Mac version stores them as data sections inside of the mach-o binary, and get loaded into memory when you run the app.

###Running the app: 1. launch the evasi0n app from finder 2. evasi0n dials out to “http://evasi0n.com/” for “ex.plistx” - if you lack a network connection, evasi0n exits out 3. checks if your language/locale settings are english or chinese, configures UI accordingly geohot’s breakdown

###Obfuscation:

Obfuscation was used on the code of the evasi0n 7 app to hide both the process of jailbreaking and exploit. While obfuscation makes code harder to read and reverse, the more important aspect of software that must be hidden is the “what” not the “how”. There are only so many ways to go about hiding what you have done without leaving traces of the tools you used for the process. One common example of this is symbol stripping, to remove clues to what functions do. However, externally linked symbols leave evidence that you cannot erase. This is how I discovered how the payloads were being stored in the binary and how to remove the “dial home” and language check features of evasi0n. There are only so many ways to accomplish these tasks, and by locating the specific calls it makes it much easier to understand what obfuscated code means in a localized area of the binary. The hardest part of reverse engineering is figuring out what something does, because by knowing the end result there is a finite number of plausible solutions that can create that specific result. I made a number of comments on the level of obfuscation of the code and the app. Namely I think the code was obfuscated well, I haven’t seen binaries quite like this before, and it was a fun challenge. The level of obfuscation on the level of the app was next to nothing. There was no attempt to hide or disguise what was going on in any way. That makes it difficult for me to accept that this was done to keep the exploit from being patched and the JB method a secret.

###The Kernel:

I don’t have much to say about the kernel exploit, I am leaving the breakdown of that to people more knowledgable experienced than myself. However what I do know is that this exploit is:

1. known (code for exploiting the kernel itself)
2. easily patched
3. not very elaborate

The full technical explanation for this should be appearing soon in the form of a breakdown post, and I shall be posting a link to that on here. To respect those involved with this I am not going to be disclosing what the kernel exploit is in any more detail.

##4. Aftermath

Overall I would say this was a very poorly executed release, rife with bad decisions made for questionable reasons. Everything about it says “sloppy”. From the exploit process itself, the binary, partnership with TaiG, reasons for release, to the stability and official standing with members of the community. Everything says it was rushed and done for unexplained reasons that border on ego and greed. I would say within the next week or two, we will see an official 7.0.5 patch come down from Apple and they will stop signing 7.0-7.0.4, potentially before a working version of evasi0n has been released to support the iPad 2 or mainstream support for the newer 64bit devices in the JB development community. The ball was dropped, big time, and there is no reason as to why they had to execute it this way. A lot of the core developers in the community were blindsided by this move, some of them are now leaving the community entirely because of the selfish decisions made for the sake of being able to release first. This needs to be recognized, by the JB community as a whole. It isn’t about who can deliver the product, because even those that never touch a line of code are still part of making the product. The product is nothing without the people that make it worth while, by blatantly ignoring them you are doing more work to destroy that ecosystem than any amount of shady business deals can do.

If anyone wants to get in touch with me to discuss anything pertaining to the evasi0n software or the jailbreak as a whole, you can find me on Freenode as “dirkg” in #evasioninvestigate, on reddit as samdmarshall, and @dirk_gently on twitter.
