# Tweaking is the new-age hacking

I am not a prolific programmer by any means. I have a programmer’s mind, and I have proven in my days I am capable of writing working solutions. I have a few tweaks in my name, and more ideas to be realized. Creating more has been about having more free time. However, my time has been spent becoming familiar with iOS-internals, because I find that I am a good learner. I have a fair understanding due to the tools we have available, made by great programmers before our time, and from documentation and examples shared by the community. Because of the nature of Cocoa and Objective-C, we can take a great adventure and introspection into the workings of third-party software, and Apple’s operating system. This provides a foundation and skills for making tweaks. We want to encourage tweak making because it has been the driving initiative behind the audience that wants to have jailbroken devices, besides for the groups that wish to only have a jailbreak for pirating apps and games. The growth of this jailbreak ecosystem has gone with the proliferation of new tweaks, ever pushing the boundaries of modification while maintaining a safe environment for the end-users.

The jailbreak development scene has given a unique opportunity to developers to express themselves in a new way. In the days before CydiaSubstrate, apps and games were not tweaked. This is a new concept; examining and debugging existing software and then rewriting portions of it with the least invasive tools available, the changes are nonpermanent and for the most part free of worry for breaking something with any lasting effect. Tweaks allow for a redefining of how software works and behaves. We do this with tweaks, and there has really been nothing like it before in the world of programming, even on the PC. There were opportunities throughout previous decades to make game patches, hacks and so forth, but it’s only with the emergence of the audience of jailbreakers and iOS that we find our unique situation. Only recently has it become feasible to make small adjustments to existing UI and modify how things work without requiring the replacement of whole parts of the code - CydiaSubstrate allows careful targeting of methods and functions.

It’s a lot of fun to discover how things work, and tweak making is the embodiment of that fun time for developers. One of the challenges for tweak making is coming up with new ideas to create, and sometimes these ideas only arise after studying the internals in some detail. If you make tweaks as a hobby, and not as a profession, you’re free to do as you wish and to focus on projects that interest you. For new tweak makers, there’re quite a lot of existing projects to learn from, but a lot of the easier projects have already been realized. Creating new original ideas that are unique is a task of being familiar with the available tweaks on Cydia, and then going to work discovering how the internal parts work, debugging and testing until you have a diagram or picture in your mind how it’s put together. When you reach a near complete understanding, you are primed to tackle whatever challenge you make for yourself.

Some of our greatest tools and resources are free: Apple’s own documentation is excellent, and for tweak makers we have a wiki and the opportunity to use class-dump to examine what methods are exposed for hooking inside the target app or process. Debugging and disassembly tools that vary from free to paid, all can be great assets for tweak makers. A well-studied programmer with some prior experience with standard projects will be in a good position to continue learning from these materials. To the contrary, a newcomer programmer, even a person with some good ideas will struggle at first with the learning curve. We recommend a core understanding of Objective-C and Cocoa principles for aspiring young tweak makers; this can be a significant investment of time, but it is really a hurdle for new tweak makers that haven’t a clue where to start. To the uninitiated, the object-oriented nature of the programming involved can be a daunting thing to realize. Generating tweak ideas can be a task for amateurs, but the writing of the code for the tweak implementation is often the result of planning and research and testing for a significant time. We find that many young new programmers are impatient because their ideas for new tweaks do take more time to materialize than they were willing to invest. Patience is a virtue of course, and the best-made tweaks are all products of careful programmers.

The greatest tweak is Activator (libactivator). Based on a commonsense idea of having more triggers system-wide, activator is also a graciously open-sourced project; the product of many months and years of work by our most senior tweak maker, Ryan Petrich. His dedication and expertise shines through in Activator, which doubles as a platform for third-parties to harness the powerful triggers from anyplace to use in their own projects. It represents a lot of research and understanding of the most obscure internals on iOS: SpringBoard and backboard. If there is one shining example to point to as a goal for a tweak maker to show how much research and careful planning can go into a tweak, that is the example to look towards. It’s a lofty project that none should consider as being trivial to do, however. For some aspiring developers it can be a great encouragement to see what is possible. Kudos to Ryan Petrich for making it, and for all he does to further the jailbreak development community.

As the repo maintainer for TheBigBoss, I have a job description for myself. Doing my job has given tremendous opportunity to be an influence or guidance for new tweak makers. Often their first experience with another member of the jailbreak development community is with myself when they first contact me or submit to the repo. We wish that all developers can be involved in the social channels of this scene: chat, forums, twitter et al., however, it’s not uncommon that some developers work in relative isolation from these social groups. My involvement then can be seen as important: I may be the only other voice that the programmer will hear, and I will give an opinion on the technical merits of new tweak projects; often this first encounter is invaluable because those developers that work in isolation are not wise to many of the caveats and conventions we hold as important in this community. Our documentation and wikis have improved to make these details more available, but still I am often the first time a developer has some interaction with someone with a greater expertise than their own. I try to give my wisdom and guidance to the developers because its in our best interest to support, if not groom, newcomer developers so they feel as part of the group of jailbreak developers, and they can be pointed towards ways to avoid some of the pitfalls that many newcomers make. I take some pride in doing this and helping in part to strengthen the developer community that is based around the tweak-making culture. I want the jailbreak platform to continue to grow and mature by the great ideas that are envisioned and the expertise to realize them.

Do not be discouraged when the task seems difficult. We have some developers with years and decades of programming experience, and we also have some with only a few weeks or months. I come from the school of thought that it should be well made and well tested, and not rushed or forced. If you have a goal, it should not be merely to have something of yours published on a Cydia repo, but to give something to the public, which will enrich their jailbreak experiences - that is for hobbyists like myself. If you have some commercial interest in Cydia, and for making an selling tweaks, do wide testing with users and alongside other tweaks to help assure a product that works for many users and their combinations of tweaks; your duty as a responsible tweak maker is to be careful while you modify the insides of others’ programs or apps, and to be thorough in testing compatibility with others’ tweaks.

Tweak making is the new-age hacking. There’re already enough reasons for you to get started with tweak development, and we need tweaks to keep the jailbreak community in bloom. Join us, learn from others, work hard, be patient, and have fun.

Optimo