# Introduction to iOS Jailbreaking. Positive and negative sides from a developers perspective

1. **What is iOS jailbreaking**
2. **How to jailbreak iOS 10 and what i need to know before jailbreaking**
3. **Post jailbreak tools:**
    a. **Cydia**
    b. **file browsers**
    c. **ssh**
    d. **appSync**
    e. **iCleaner**
4. **- Future of jailbreaking**
    a. **Apple bug bounty program**

## What is iOS Jailbreaking ?

Similarly to it's desktop sibling macOS, Apple's mobile operating system iOS is build around a Unix core. Unix based operating systems have a lot of things in common. One very vibrant example of such similarity would be that all of them share identical role based security models, whose having one special role called "root". This special role, sometimes also reffered as "superuser", has the permission to do pretty much everything with the operating system and it's one of the most secretly kept things in iOS.

Let's look at other permissions from the iOS role based security model. I am not going to dive in great details here but in general some of them form a group called "**Seatbelt (Sandbox) profile**", used for all apps inside /var/mobile/Applications (up to iOS 7.x) or /var/mobile/Containers (since iOS 8) directory i.e. AppStore apps. Basically seatbelt is a unix kernel extension which runs on the iDevices as a part of the operating system, whose responsible to restrict one's app to reach resources that are not in it's own so called "container", for example access memory areas used by other applications, access to directories inside filesystem containing data from other apps or system files. This kind of sandboxing mechanism is what defines iOS as the leading mobile operating system regarding security of user data.

Let's get back to the question. What exactly is iOS Jailbreak ? Which iDevice we can refer to as jailbroken ?
To **achieve a Jailbreak on a device running some version of iOS, means to find a way around sandboxing mechanism of iOS - "Seatbelt"**, so that your software could have access to resources that were not provided for use by it. Up until now, jailbreak of iOS has been achieved by group of software exploits found inside iOS, which at the time present Apple, as a product owner, was unaware of their existence.

To be jailbroken does not exactly mean to be able to run code with privileges from the "root" role, but some jailbreak methods can achieve that and grant you such access. In short this means that you can alter or modify execution of all of software ran on the jailbroken device, leaving no options for software that being modified to notice this. Sounds interesting? Let's cut to the chase by showing you how to jailbreak your device and then preparing the environment to do some cool stuff.

## How to Jailbreak iOS 10 ?

But wait, lets first answer some questions that maybe most of you are asking at this point:

- **Is it safe?**
- **Can i make a irreversible damage to my iOS device ?**

The former i would answer with it depends on the answer of the latter and here is why - the process of jailbreaking itself is safe, but successfully achieved jailbreak opens a lot of possibilities to mess things up with your device, so be careful. You have been warned.

- **What exactly can you mess up after successful jailbreak ?**

Let's point some examples here - you can alter the boot sequence which will render you device unbootable, you can hook some system method and add code to them which have the potential of throwing exception leading to crash of the given software. The latter may sound less scary if you think in context of apps, after all you remove the specific app and your code additions, and everything is fine again. But what if you hooked some method from the SpringBoard ? If you manage to crash SpringBoard, that means you have no way fixing your device other than, restoring it, if you haven't prepared your ground for that earlier. The examples that i gave you until now are all fixable by full iTunes restore. But on the other hand there are some things that would led to hardware failures which means IRREVERSIBLE damage to the device, like for example adjusting the power output of cellular radio, adjusting the charging rates to values way beyond those set by Apple, however up until now, i have never heard

of someone actually doing such harm to device unpurposely, which is a sign that it's really really hard to be achieved, especially by jailbreak newbies.

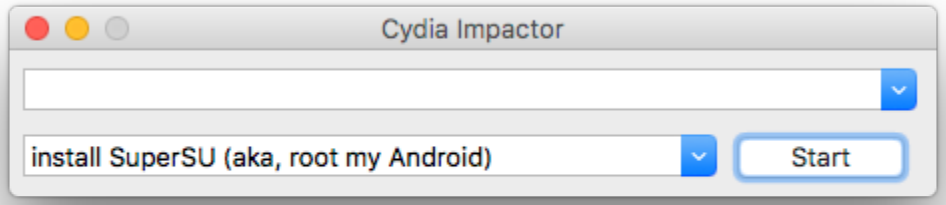
Generally speaking, there are **two types of jailbreak - tethered and untethered**. The former, which is not permanent, needs to be done after each reboot because you lose it after restart. The latter is permanent which means that your device will be jailbroken after reboot.

## How to un-jailbreak ?

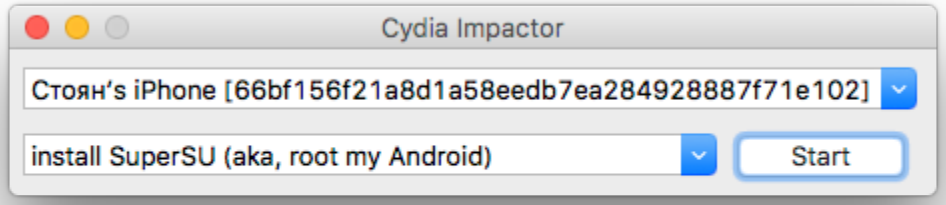
If you have untethered jailbreak you have to restore your device using iTunes. Do not use the reset options under Settings > General > Reset, because it will not finish successfully, leaving your device in recovery mode. But on the other hand if you have a tethered jailbreak, then you just need to restart your device and you are unjailbroken, however there are still some files around your phone from which someone could track that your phone has been jailbroken.

So in both cases the convenient way would be to restore your device using iTunes but there is one catch here - **iTunes only allows installation of latest iOS release at that time**, so you may be left with iOS version which can't be jailbroken, or if you have previously saved your shsh blobs for specific iOS version you can install it also with the help of one tool called "Prometheus".

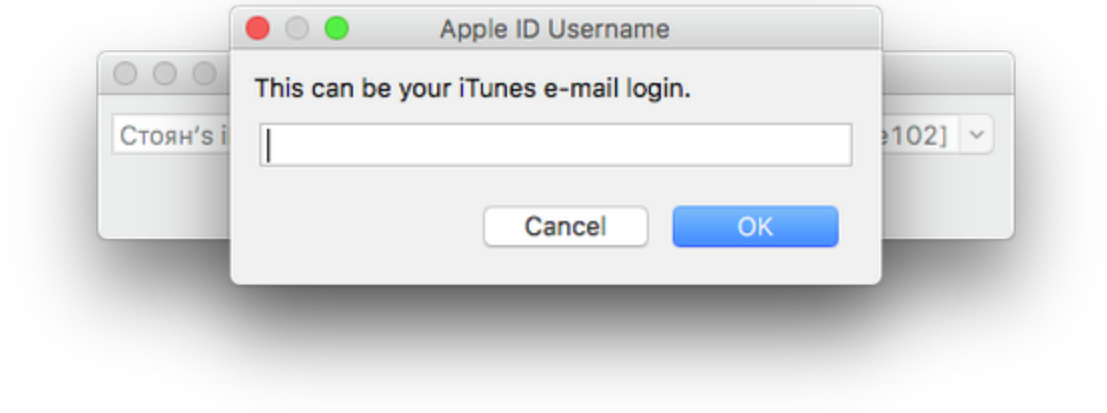
So let's have a look at one **method of jailbreaking iOS 10 - "yalu102"**. At this point yalu102 is tethered jailbreak, meaning that you have to rerun it every time you reboot your device. Basically this is app which you can not find in App Store, but you can download it's source from developer's github repo https://github.com/kpwn/yalu102 , and compile it yourself if you are knowledged enough to fix the dependency issues. In case you are not, there are pre-build ipa's of the app. You can obtain them from links pointed inside github page, or from the official yalu102 website - https://yalu.qwertyoruiop.com . After obtaining the compiled application, you should somehow sign it in order to run it on your device. If you are compiling it yourself, you could set this up in the traditional way in XCode, but if you go the other way of getting precompiled version you may use a piece of software available for both MAC's and Windows PC's, called "CydiaImpactor" or use some other approach. The purpose of this tool is to code sign your version of yaly102, so that you could run it on your device. You just launch the tool,




connect your device to the computer,

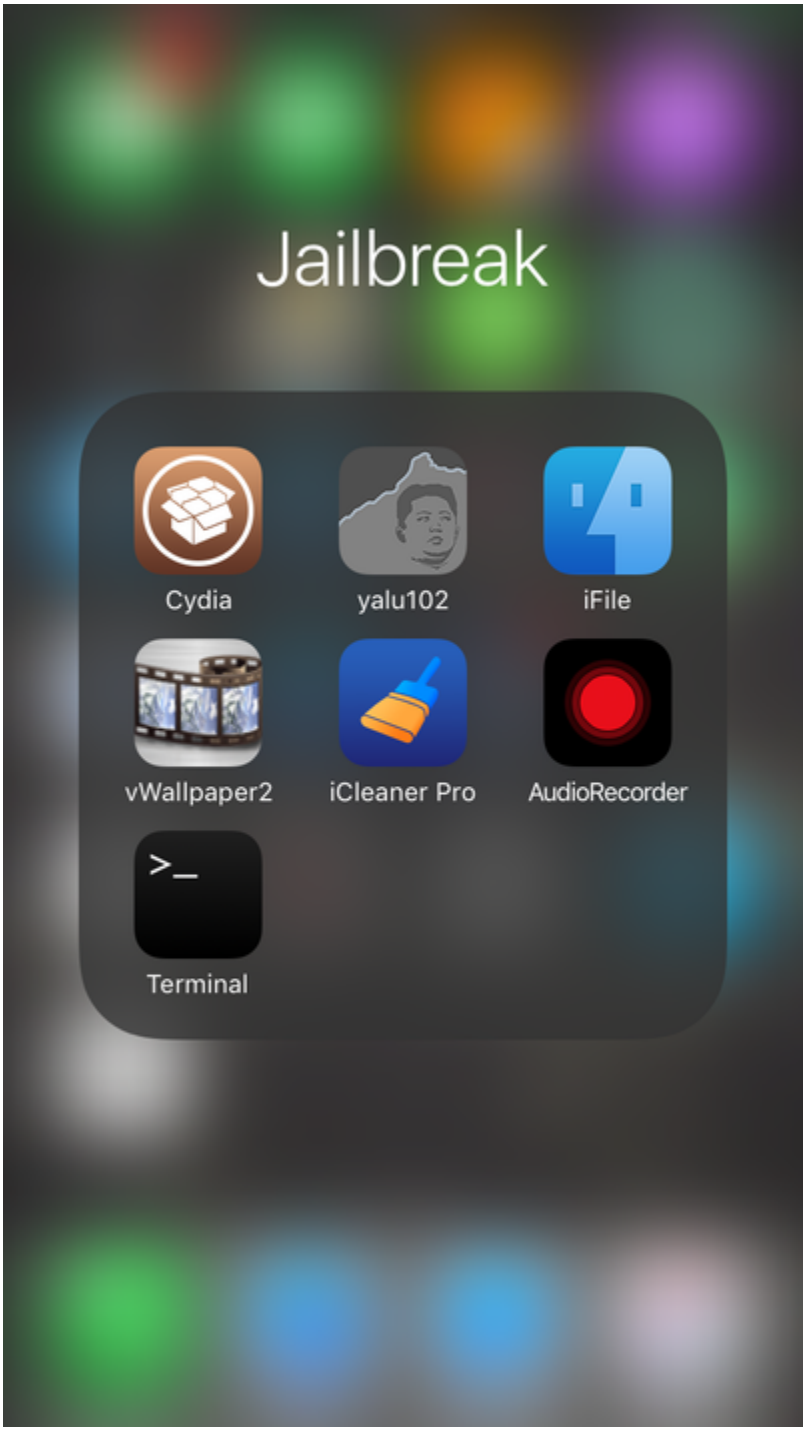



drag and drop your version of **"yalu102.ipa"**, and type in some Apple ID which would be used to create and hold the provisioning profile used to run yalu102 on your device, and you should see yalu102 app on your spring board.




At this point you are ready to go! **Launch the app, and press button on the center of the screen saying "go"**. If you are having troubles launching the app go to System Settings > General > Profiles & Devices Manegment and trust the developer with the email you have entered in CydiaImpactor. After a successful launch you should see your device **re-springing - that is a term in jailbreak world which means restarting SpringBoard process**. When everything is done you will be left at your Lock Screen. Unlock your device and search for a newly installed app

called "Cydia". If "Cydia" is present, congratulations, you are now jailbroken!




Now let take a look at some really useful tools which most of the jailbreak users are using. The first on the list would be **Cydia**. We can think of Cydia as the app store of jailbroken devices. In it you can find pretty much every piece of software written specifically to run on jailbroken devices, like for example - all sorts of visual tweaks, command line tools, compilers(yes you read that right you can compile code on your iDevice and debug or run it directly on it), various retro game consoles emulators, file browsers and anything else that comes to your mind. You can even build your own tweak and publish it on some of the many Cydia repositories, or even create your own repository publish your software on it and share it with anybody for free.

Next on the list would be **iFile**. It's basically "Finder" for iOS. It let's you browse the filesystem of your device with privileges dependent on the type of jailbreak you have, in our case of yalu102, that would be root access. **iFile has image and video viewers, text editors, pdf viewer, office docs viewers and it even has this really nice feature that hosts website in your local network, and from that site you can very easily upload or download files to your device.** Some very interesting directory to look at if you are iOS developer is **"/var/container/Bundle/Applica**

**tion"**. In this directory are stored all of the apps you downloaded from the app store to your device. You can observe their resources - sound files, images, assets, storyboards, nib files, all sorts of .plist files including "info.plist", json's, xml's. **That is actually the "Main Bundle" of the application.** Furthermore, **if you are observing a Hybrid app, you can actually see all of it's htmls and alter them as you wish.**

The next tweak that i am going to present to you is called **"OpenSSH"**. When installed **this tweak lets you open ssh remote session** into your device from your local network and again you can upload and download files under SFTP(ssh file transfer protocol). Other than that you can issue commands from the command line as with any other ssh session and it supports multiple sessions at the same time. Unfortunately OpenSSH is still in progress of updating in order to support iOS 10, but there is an alternative developed from the creator of yalu102 jailbreak - **"Dropbear"**. It creates ssh tunnel over usb, but in the end it can be used in place of OpenSSH.

Jailbreak also introduces other way more darker features that Apple are fighting really hard against, one of them would be the ability to install cracked versions of apps, both paid and free. Until iOS 5 there was even a cracked app store called "Installous", whose responsible for the bad reputation of iOS jailbreaking. Nowadays there is a tweak called **"AppSync"**, which let you install all sorts of apps, on your device trough iTunes, without the need to code sign them, **violating strongly app distribution rules**. On the other hand there is a tweak which again you can install from Cydia called "**AppAdmin**". This tweak lets you install older versions of app right from the app store. So if you want to test something on older version of your app, you can do this very easily, or if you didn't like the latest build of the Facebook app, you can easily downgrade to the version you was previously on.

The last tweak that i am going to mention in this post is **iCleaner**. It's purpose is to clean the operating system from temp files, log files, crash reports and all sorts of junk files. **iCleaner also has the ability to stop or start system daemons, which let you tune the performance of your device** and can help you make it run smoother. For example you may never use the iOS Game Center, you can safely switch it off, or maybe you are sick of your device showing you available updates, just switch their daemons off and they are a thing of the past. Very useful tweak! You should definitely try it if you have jailbroken your device.

## The future of jailbreaking is really unpredictable.

Apple are coming up constantly with fixes for used exploits, making their ecosystem even more secure. Furthermore, in September 2016 Apple had launched their so called **"Bug Bounty Program"** whose purpose is to get security researchers interested in iOS. The Bug Bounty program offers really appealing prices:

**Secure boot firmware: $200,000**
**Extraction of confidential material protected by the Secure Enclave Processor: $100,000**
**Execution of arbitrary code w/kernel privs: $50,000**
**Unauthorized access to iCloud account data on Apple Servers: $50,000**
**Access from a sandboxed process to user data outside of that sandbox: $25,000**

(resource data can be found on here)

which would divide the whole community of jailbreak developers in two sides, the former - would be developers who are staying true to the jailbreak scene and continue to make public releases of jailbreak methods, and the latter would consists of those developers who would rather get the prize from apple and report the vulnerabilities to then, leaving the jailbreak community behind.

In conclusion iOS Jailbreaking, for me is something that every iOS Developer should try. It is not sure if it would be helpful for them, but at minimum it will widen their view point of the operating system they are developing for.