# Evading the Sandbox

As a security measure and to keep apps on the device from sharing data or interfering with each other, iOS includes a security system known as the sandbox. The sandbox blocks access to files, network sockets, bootstrap service names, and the ability to spawn subprocesses. Part of the jailbreaking process involves modifying the sandbox so that all processes can load Cydia Substrate, but much of the sandbox is left intact to respect the security and privacy of the user’s data.

With each new release, Apple further improves the sandbox to improve privacy and security. When building extensions or tweaks that need to share information across processes or persist data to disk, this can be restrictive. One approach is to survey the sandbox restrictions that exist on the processes where the extension is to be run, and choose file paths and names based on them. This is common, but can leave oneself stranded when Apple tightens the tourniquet and as of iOS 8 there is no location that all processes can read and write successfully. A better approach is to do all of the interesting work inside a privileged process such as SpringBoard, backboardd or even a manually created launch daemon of your own. Child processes can then send work to the privileged service. This ensures that as the sandbox tightens, your extension will still behave properly as long as it can communicate with the service.

Oddly enough, as of iOS 8 Apple has also decided to limit which services an app store process may query. This makes nearly all forms of inter-process communication ineffective on iOS, outside of the well-defined static services that Apple has designated. RocketBootstrap was created as a way around this that simultaneously allows additional services to be registered and respects the security and privacy of the user’s data. Services registered with RocketBootstrap are made globally accessible even in spite of very restrictive sandbox rules and it will serve as a single project that needs updating as the rules change.

Ryan Petrich