DiagBootX

DuffyAPP-IT
@j_duffy01 & @_JaccRob

Contents

DiagBootX Application

Purpose

DiagBootX is a tool that was developed in order to automate the process of sending unsigned boot images to the iPhone, as well other functions including automatically configuring serial connections (requiring an un-named serial cable with a lighting connector on the end..) and interacting with the device via the **serial port**. This tool was not developed to make the impossible possible - It was developed to automate tedious manual processes.

This will allow for inspection of the device hardware on a low level, and testing of bypasses for services such as **FMI** and **Device Activation**, by triggering the device to deactivate as a result of modifying the stored **serial number**.

Requirements

The software has been compiled to run on macOS Catalina and above.

The code, though written in an Objective-C wrapper for the GUI, has it's core functionality written in C which is compatible with other platforms and operating system versions. By modifying the info.plist, I was able to launch the GUI on macOS High Sierra.

In a future release, I will recompile for a lower minimum operating system version in order to maximise the amount of compatible machines. The application does not utilise any API's or features specific to newer macOS versions, so there should not be any issues with this process.

Alternatively, the CLI binary is also available, compiled for macOS. I will compile this for Windows and Linux devices soon.

In addition to this, the user should have a fork of **IPWNDFU** to allow for their specific device to have the bootloader patched. This is to allow a custom **iBEC** and **iBSS** to be uploaded and booted on the device. This is critical for the aim of this application to be fulfilled.

The user must also possess a pre-patched iBSS and iBEC in order to boot to the specific image/ partition as per the user requirements.

Example Usage Scenario

"User Requests A New Serial To Be Written To Device"

Using the example of Apple DIAGS, the **diags.img4**, **iBSS.img4**, **iBEC.img4** would be placed in the **Assets** directory, and when ready, executed automatically. The user will then have the possibility to interface with **DIAGS** via the **DCSD serial connection**. It will then be possible to use the **SN** command to write a new serial number to the device.

I plan to create a scripting language which will allow the user to easily select values to configure a set of devices with, which could enable repair businesses to automate certain tasks easily.

Usage Possibilities

This software can be utilised for a variety of purposes. As of now, select functionality has been released. This includes booting unsigned images, utilising IPWNDFU and using a pre-patched iBSS and iBEC.

I am currently testing functionality which would result in an easily learnt scripting language to modify low level values of an iOS device, such as the Serial Number or restrictions in relation to Device Voltage etc..

In the future I am hoping to implement automatic patching of the iBSS and iBEC to fit the users use case, and to automatically establish a serial connection to the device.

Part of serial connection functionality would include automatically determining the device baud-rate using an inbuilt function.

Development Process Overview

https://www.youtube.com/watch?v=6_8cY4NWxxs - Public Resource