PATCHING APPLICATIONS FROM APPLE'S APPSTORE WITH ADDITIONAL PROTECTION

Author: Reilly

Editor: Shub-Nigurrath

ARTeam

FOREWORDS

Since Apple opened the AppStore tens of thousands of application are available for the iPhone and the iPod Touch and it keeps growing. All available apps are protected by Apple's own DRM system called Fairplay. The binaries are encrypted.

From the jailbreak of the iPhone it only took a short time till Fairplay was broken. An iPhone port of GDB made it easy to crack those apps by dumping the decrypted binary from the iPhones RAM http://blog.gauravgiri.com/2008/07/drm-fail/.

After this "breakthrough" a few CLI scripts (xCrack, DCrypt) were created which semi-automated the cracking process. But this was nothing compared to Crackulous, which offers a GUI for automatically cracking bought Apps and making it easy to distribute to all the pirates.

Of course the developers are aware of this and some of them are trying to protect their applications with their own methods. They check for the modifications done to the package because they are not allowed to use serials or other methods to protect their work.

This tutorial focuses on finding and disabling these checks. It is heavily based on Shub-Nigurath's "Primer on Reversing Jailbroken iPhone Native Applications" which offers a great introduction on the Mach-O file format and the Objective C programming model and how IDA can be used to disassemble those files.

1 TABLE OF CONTENTS

DISCLAIMER/LICENSE

All code included with this tutorial is free to use and modify; we only ask that you mention where you found it. This eZine is also free to distribute in its current unaltered form, with all the included supplements.

We have potentially illegal stuff inside. All the commercial programs used within our tutorials have been used only for the purpose of demonstrating the theories and methods described. These documents are released under the license of not using the information inside them to attack systems of programs for piracy. If you do it will be against our rules. No distribution of patched applications has been done under any media or host. The applications used were most of the times already been patched by other fellows, and cracked versions were available since a lot of time. ARTeam or the authors of the papers shouldn't be considered responsible for damages to the companies holding rights on those programs. The scope of this document as well as any other ARTeam tutorial is of sharing knowledge and teaching how to patch applications, how to bypass protections and generally speaking how to improve the RCE art. We are not releasing any cracked application. We are not at all encouraging people to release cracked applications; damages if there will be any have to be claimed to persons badly using information, not under our license.

This disclaimer applies to all ARTeam releases and tutorials!

VERIFICATION

ARTeam.esfv can be opened in the ARTeamESFVChecker to verify all files have been released by ARTeam and are unaltered. The ARTeamESFVChecker can be obtained in the release section of the ARTeam site: http://releases.accessroot.com

2 BASICS

In this part we are going to learn about the tools needed for reversing and patching, the file structure of those apps, some basic things about ARM opcode and after that about the modifications, which are done to the application package while the cracking process.

2.1 TOOLS

This is a short list of needed tools.

2.1.1 JAILBROKEN IPHONE OR IPOD TOUCH

Maybe the most important tool/gadget is a jailbroken iDevice (or also called iAwesome ;)) with iPhoneOS 2.x. It has to be jailbroken to run modified apps on it. Tutorials on how to jailbreak and other iPhone specific stuff can be found at iClarified.

2.1.2 IDA 5.2 OR NEWER

Since version 5.2 IDA's support for disassembling iPhone applications (Mach-O and ARM) has greatly improved. Earlier versions have problems recognizing the instructions. However even the newest version sometimes has problems handling thumb opcode. It seems that developers know that and are trying to use this to protect their applications from being disassembled.

Note: For Mac users it's highly recommended to run IDA in virtualized Windows (VMWare Fusion or Parallels) and not to use the native OS X app.

2.1.3 HEX EDITOR

Without a descend Hex Editor you won't be able to modify the binaries. For Mac users, I recommend 0xED. For Windows, you can use tools like HexEdit

2.1.4 SFTP/SSH CLIENT

Use a SFTP/SSH client to access your iDevice's file system and execute commands. Cyberduck and WinSCP are recommended for Mac OS X and Windows respectively.

2.1.5 TOOLS ON YOUR iDEVICE

There are Tools you need to have on your iDevice in order to run and/or sign the modified binaries. All are available via Cydia. For Crackulous and the MobileInstallation Patch (MI) you have to add a repository like http://cydia.hackulo.us.

a) Crackulous: to crack bought apps. This is a very easy way to obtain the decrypted binary. The Application will be packed in an .ipa (iPhone/iPod Touch Application) file which can be opened with any Zip-Tools.
b) MobileInstallation Patch (MI): to install cracked apps. Without this you won't be able to install Apps from their ipa container.
c) MobileTerminal: A nice Terminal to execute commands.
d) OpenSSH: to establish an SFTP/SSH connection to your iDevice.
e) Link Identity Editor (ldid): to sign the patched binaries. Without signing the modified binaries won't run. Signing can be done using the ldid –s /PathToYourApp/binaryname command.

Note
.ipa files are essentially zipped/compressed archive files utilizing an Apple specific file extension, by changing the extension to .ZIP and unzipping in a program such as WinZip - the files contents can be accessed.

2.2 THE FILE STRUCTURE OF THE APPLICATIONS

iTunes uses container files with the .ipa as extension to install the applications. Those files can be opened and edited with any Zip-Tool. Inside you will find iTunesArtwork, which is used to create the icon in iTunes and a X.app which contains the application files.

On your iDevice, the applications get installed to a folder called /User/Applications/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ (where X stands for characters). If you want you can use AppLinks from Cydia, which will automatically create shortcuts for all AppStore applications to provide faster access to them.

Inside this folder you'll find the application package X.app and several folder where the application can store it's data in.