CERIAS

The Center for Education and Research in Information Assurance and Security

Exploring Memory Forensics on iOS Devices

Colin Cowie

Marcus Thompson

Department of Computer and Information Technology

Abstract

The increase in iOS devices has made it a bigger target for malware. Currently, static analysis is the primary tool used for analyzing iOS malware. This research will investigate the potential of using Random Access Memory (RAM) to help analyze malware on iOS 10.1 devices.

Problem Overview

As of 2016, Apples iOS devices make up over 15% of the smartphone market worldwide (IDC, 2016). Despite Apple's iOS being regarded as “one of the most secure operating systems”, with a growing marketplace, the amount of threats increases too. Since July 2012 when Kaspersky Labs discovered the first iOS malware, the public has seen a steadily increasing amount of iOS malware. The primary method for analyzing iOS malware currently is through statically reverse engineering the binary of the application. Analyzing Random Access Memory (RAM) is a technique that digital forensics incident responders use to find critical data that was previously undiscovered through the use of traditional forensics techniques (Case, 2014). The use of memory forensics is gaining popularity and highly beneficial. At the time of this study (2017) there is no standard for performing memory forensics on iOS devices. This research aims to determine if memory forensics could potentially be a useful technique for iOS forensics and its potential for gaining insight on iOS malware.

References

Case, A., Ligh, M. H., Levy, J., & Walters, A. (2014). *The art of memory forensics: detecting malware and threats in Windows, Linux, and Mac memory*. Indianapolis, IN: Wiley.

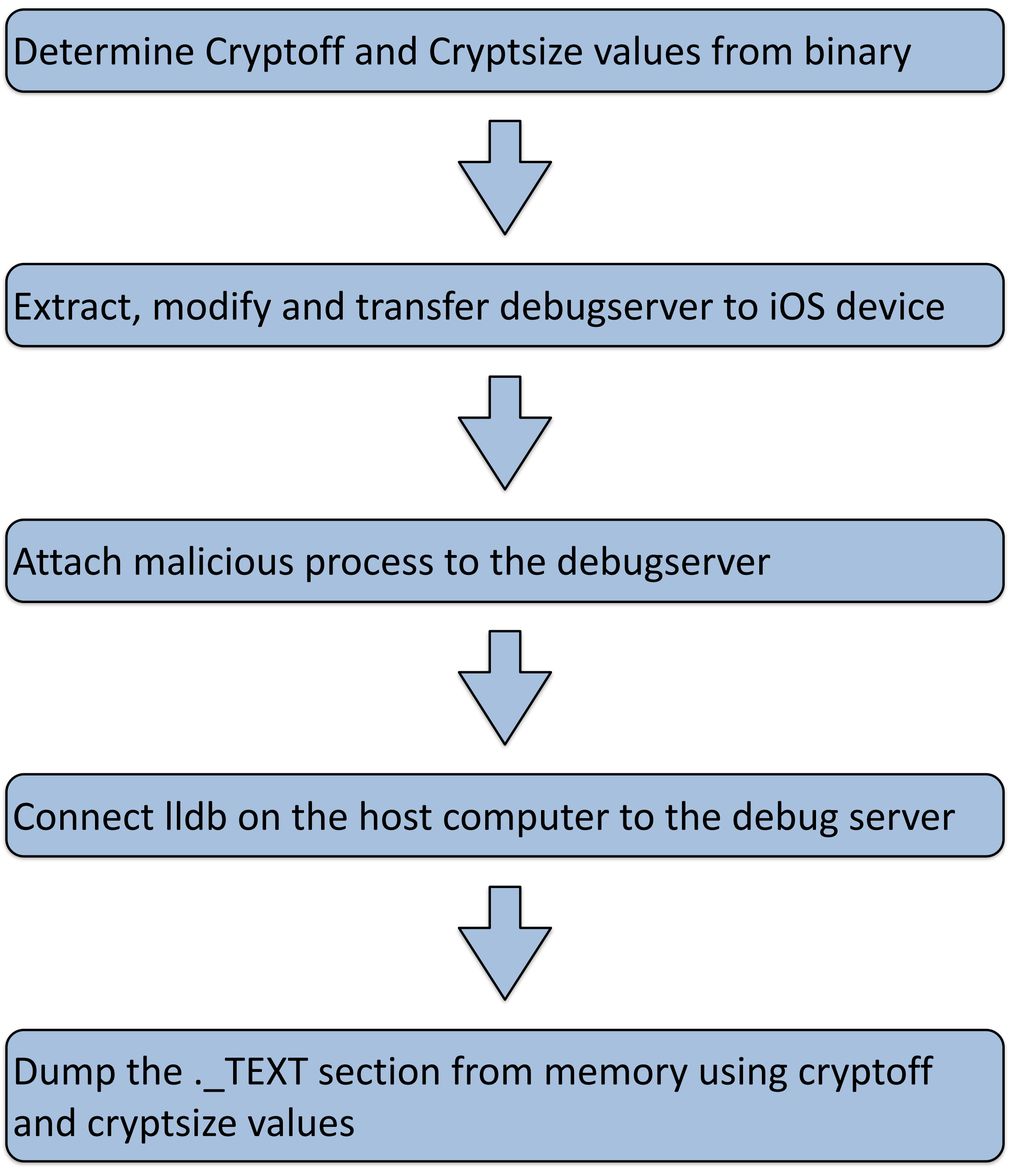
IDC: Smartphone OS Market Share. (2016, October). Retrieved February 10, 2017, from http://www.idc.com/promo/smartphone-market-share/os

Methodology

For this study iOS malware from the following families were attempted to be dumped from memory:

- KeyRaider
- Muda
- TinyV

Debugserver is a console application that acts as server for remote gdb or lldb debugging. Debugserver comes as a part of Xcode. Extracting and modifying the debugserver binary allows for remote debugging processes on iOS devices. The ._TEXT section of the process contains the code being executed. This sections is encrypted but with the cryptoff (file offset of encrypted range) and the cryptsize (file size of the encrypted range) you can calculate the decrypted location and dump this section using lldb.



CERIAS

PURDUE UNIVERSITY