Writeup

Kevin Turkington

I. Overview

In Alex Hinchliffe and Fernando Ruiz lectures, Fernando gives an overview of mobile security and exploitation of the the android kernel. For the exploitation of the android operating system Fernando gives an explanation of the different malwares associated with both the Android and IOS platforms. Specifically the various ways hackers create and obfuscate their exploits as well as exploit infected users for financial gain. A great example he gives is utilizing premium rate phone numbers to force infected users to be charged for messaging said number. In addition to querying users installed apps and replacing specific banking apps for more malicious ones to gain login information.

II. Attack Vectors and Methods

The primary attack vectors for mobile applications is to gain permissions to specific functionality of the users phone this can be anything from simply having the ability to access the internet to having read and write (sending) access to SMS messages. To do this the malware create must add specific privileges to the android manifest file within their payload application. This can be problematic for the malware developer because to gain these permissions the user must allow them. However, this is not the case for rooted or jail broken devices, because any security features provided by the kernel developer can be disabled. Because of the elevated privileges that come with rooting a device.

Another vector susceptible to malicious intent is the use of reflection in the Java programming language. This involves altering core functionality of a segment of code during its run time, this exploit takes advantage of Javas nature of being a partially compiled and partially interpreted language. Java reflection as an overview is invoking calls of another class through the use of the `.invoke()` method. Another reason why reflection is used is because it makes investigation of a malicious programming extremely difficult to analyze for any mobile malware analyst.

III. Static and Dynamic Analysis

Analysis of a malicious app can be done from a set of tools similar to those presented in week one and two:

- Android Virtual Device
- Androgaurd
- Dex2Jar
- APK-tool

- JS-Gui

for static analysis Android studio provides a java decompiler to developers and analyst for looking at the source code of a application. However if the application has already been packaged into an APK, the Dex and xml files can be extracted and analyzed using thw Dex2Jar tool. However as previously stated java reflection and native JavaScript interpretation can make most analysis methods difficult. Leaving the researching with the option of decompiling the APK itself into assembly to understand how it works on the most primative level. Another option is using an Android Virtual Device as a goat to infect and analyze the functionality of applications during run time. During run time analysts can find out if the goat had any messages attempt to be sent out to outside numbers, if any external communication to servers were made, or if any previously installed applications were replaced by new ones.