OREGON STATE UNIVERSITY

CS 373

WINTER 2019

Week 9 and 10

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath

I. Overview

This week we will be diving deeper in to mobile security. As we go through the lectures I will talk about sections of the lecture that I found interesting or insightful. I will also list out terms that I learned through out lecture this week.

II. Lecture Topics

A. History of Mobile networks

First call using a cell phone was in April 3, 1973. I find this striking as this wasn't all that long ago that we didn't have cell phones. The first system that was used to send cell phone calls was 1g. 1g was bad. It didn't have many features and it had no security. The next generation of the cell phone network was 2g which was much better and implemented many more security features. Next was 2.5 G which allowed data to be transmitted of the cellar network. after 2g there was 3g which mostly implemented speed enhancements as well as 4g or LTE.

B. history of mobile OS

Mobile OS has an interesting history. Early on there were many players trying to get into the mobile OS world. The mobile OS world really started with a few key players. Palm, the people who made palm pilots, Microsoft, Symbian and RIM (black berry OS). One thing I find really interesting is that none of those players are still major players in the mobile market today. The big game changer that caused all of these other companies to become less popular was IOS. It is interesting to note however that IOS didn't capture the majority of the market share but rather caused a revolution in the way that people use smart phones. The other major player is google. Goolge actually started playing around with the idea of mobile OS before IOS was released however, IOS beat android to the market. Today however android has the majority of the market share.

C. Mobile OS Security Features

In this section we will only be talking about the big 3 players in mobile OS today. Android, IOS and Windows. The different platforms have different security features. Android allows any one to create apps and will auto approve any app submitted to the app store. The registration costs for the app store is low. This allows for a divers pool of apps but also allows for apps that shouldn't be downloaded to slip through. The app permissions are managed statically. App isolation is done through sand-boxing and through permissions. Another interesting thing about android is that it allows third party installs.

iOS on the other hand takes a much heavier hand in how apps are approved and distributed. IOS does not allow its apps to be deployed with out approval. The signing process is managed by Apple. The fee to use the public app store is 99 dollars per year. App permissions are dynamic meaning when they are used. It's also important to note that 3rd party installations are not allowed.

Windows is much the same as IOS with only a few differences. The barrier to entry on Windows is lower much lower, 19 dollars. My guess would be because they just want people to write apps for windows phones. The app signing is also managed by windows much like Apple. The windows phones had a slightly different strategy for App isolation though. The windows approach was to have different permission levels for each type of app. The kernel and drivers have one permission level. System apps have another permission level and standard apps have another permission level.

D. Security Bypasses

There are several security bypasses for Android and iOS. In iOS the main security bypass is actually done by the user on purpose. This is done to run software or code that is not authorized by apple. This is called jail breaking. This can also be done to Unlock carrier-locked IPhones. Some times this is also done by attackers to take advantage of exploits in mobile browsers.

The other main exploit that exists today is rooting in Android. The purpose of this exploit is to get root privileges on that phone. This is much the same as exploits on PC's and other linux machines. This can also be done to bypass restrictions from manufactures and carriers. This also allows attackers to do what ever they would like with people's smart phones.

E. Modern Mobile Hacks

Initially, there were not that much malware written for mobile. In this case we are talking about the period of time around the smart phone revolution (2008 - 2010). This is referred to as the calm before the storm. This calm was ended because these operating systems were starting to use a bunch of new features such as, downloading files for the internet and polymorphisim. One of the first mobile attacks was a botnet that attacked Symbian. This worm was called YXES. This was spread via a URL sent through SMS. This was also one of the first pieces of malware that was signed using a valid digital certificate. Another piece of malware that was the first to the game was IKEE. This was the first IOS based malware. This was first discovered in the Netherlands in 2009. The attack worked by Jail breaking the iPhones and putting ransom ware on the IPhones. Several other variants of the same malware were found in different locations around the same time.

There are also several other notable attacks are FakePlater and Tapsnake. FakePlayer was a very limited function media player application. This application pretends to be a media player while sending SMS messages to premium rate numbers. This was done in the background with out the users consent. Tapsnake was found at a similar time as FakePlayer and was very similar. However, Tapsnake tracked GPS coordinates and sent them to a remote server. This application was actually served through the Android Market (google play). This attack also set it self up in the boot list and started right after boot.

F. Malware Revolution

We notice that during 2010 there are very few attacks on android phones. However, by 2011 there were many more incidences of malware being reported. Geinimi was the first android botnet. It was first found in December 2010 after Christmas. It was mostly distributed in third party markets in China. This application leaked sensitive data to a private server. This malware also installed other malware and prompted the user to uninstall an application but instead executed commands send by a remote server. As far as malware goes Genimi was fairly sophisticated.

One of the worst things to happen for the android market was DroidDream. This was considered the Android market nightmare. This malware was actually found by a redit user. 21 apps on the Android market were repacked by Myournet and 50 packages were repackaged with DriodDream. This was done using simple XOR obfuscation or an embeded key. This type of malware would attempt to root the device. If the root is installed it would hide it self as an sqlite.db file and download more applications. This was the first malware to be found in the android market and had 50k to 200k installs in only four days.

DroidDream caused serious backlash from google. Google removed all malicious apps, suspended malicious developer accounts and added automatic security update patches. However, Malware authors took advantage of the fix by repackaging the Android security tool package with malicious code.

G. Android Architecture

The android system takes advantage of a layered architecture. The first layer is the Linux kernel. The second layer are the android libraries. At about the same layer but separated from the libraries are the android run-time environment. Next is the application framework layer. The final layer is the applications them selves.

The Android operating system is designed for devices with limited processing power and memory. It is designed with a sand boxed environment for security and performance. It is a register based VM which preforms better than a stack based VM. Android uses the Dalvik Executable format or dex.