Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.263 – 268

RESEARCH ARTICLE

An Assess Android Antimalware that Detects Malicious Dynamic Code in Apps

Miss. Srushti Hatwar[1], Prof. Chetan Shelke[2]

[1]CSE Department, PRPCE, SGB Amravati University, India
[2]CSE Department, PRPCE, SGB Amravati University, India
[1] srushtihatwar24@gmail.com; [2] chetanshelke7@gmail.com

Abstract **--Android is currently the most popular operating system and a considerable number of Smartphone's, tablet computers ship with Android. However, users feel their private information at threat, facing a rapidly increasing number of malware for Android which significantly exceeds that of other platforms. Antimalware's software promises to effectively protect against malware on Smartphone's and many products are available for free or at reasonable prices. We systematically analyze the security implications of the ability to load malicious dynamic code in Android apps. We assess an Android Antimalware software tool to detect attempts to load malicious code and from the study of many online applications we observed, that malicious code is loaded in an unprotected way is a major issue. We also show how malware can use code-loading techniques to avoid detection by exploiting a conceptual weakness in current Android malware protection.**

Keywords **-- Antimalware, malicious code, malware, Android, Smartphone**

I. INTRODUCTION

Current years have seen the Android platform gain more and more fame and a considerable number of Smartphone's, tablet computers ship with Android. Specifically, Google announced at its developer conference Google I/O in May 2013 that 900 million Android installations have been activated since the launch of the system in 2008[1]. The large number of devices running on Android operating system provides great opportunities for developers to develop various applications on android platform rather than developing on single platform. Unfortunately, the same economic incentives appeal to criminals as well. By targeting Android, they have the chance to perform malicious activity on millions of devices. The most essential factor for Android in making it an attractive target for cybercrime is the opportunity to install apps from unknown sources, which makes it simple to trap users into installing malicious apps. Accordingly, the amount of malware for and attacks touching Android is progressively growing. For example, malicious applications take users' private information or use cost-sensitive functionality such as premium SMS to generate revenue for the attackers [1]

II. BACKGROUND

In this chapter, we will analyze claims concerning the malware threat on the Android platform and the security level offered by Android antimalware, as reported by antimalware vendors and independent antimalware tests.

In the following, we will focus on the most important virus channels for typical malware which can be found in the today. For more in detail information on Android spread channels, persistent infection, and malware methodology in general.

The official Android app market is somewhat well controlled. While techniques exist to circumvent this, the Google Bouncer dynamic heuristic malware detection service exists to protect the official Android market, called Google Play [4]. In order to counter the spread of malicious content in Android's main application store, the Google Play store, Google introduced a vetting mechanism for applications in 2012 [3]. Google employees also have the option to manually take off malicious apps from the market and even remotely clean it from devices. Pirated malware gets removed fairly quickly and well known and straightforwardly measurable malware does not get admitted to the Google Play Store at all.

An important way for malware to circumvent the Bouncer is by loading external code at runtime. For example, imagine an app that pretends to be a game. During the Bouncer's analysis, the application does not expose any malicious activities, and even its code does not include any malicious functionality. But once the application has been approved by the Bouncer, admitted to the store and installed by users, it starts to download and execute additional code that performs destructive activities. We show that using this technique, applications are able to avoid detection by the Bouncer and several anti-virus products.

A. Android

Android is an operating system for portable devices such as Smartphone's and tablets. It controls to protect user data, system resources and provide application separation. It is based on the Linux kernel. Android applications are generally written in Java and compiled to Dalvik bytecode. It provides the choice to run so-called native code, i.e., machine code for the device's processor. The general way of invoking native code on Android is through the Java Native Interface (JNI), a uniform interface for interaction between Java and native code. At the system level, loading native code means that Dalvik, Android's virtual machine, loads a Linux mutual object and allows Java code to make calls to the contained native functions (and vice versa). Native code runs in the equal sandbox as Java code. Specifically, the same permissions are imposed on native code as on Java code. As an alternative of running Java bytecode, Android runs Dalvik bytecode, which is formed from Java bytecode. In Dalvik, instead of having various .class files as in the case of Java, all the classes are packed together in a solitary .dex file. Android applications are made of four types of components, namely activities, services, broadcast receivers, and content providers. These application components are implemented as classes in application code and are stated in the Android- Manifest. The Android middleware interacts with the application through these components.

III. ANALYSIS OF CODE LOADING

In this section, we first observe different techniques that applications can use to load exterior code at runtime, and briefly outline why they can cause protection issues. We then explain reasons for benign applications to load malicious code.

We recognized a number of ways to let Android load additional code on behalf of an application. These techniques can be used by malware to evade detection, and improper exploit can make benign applications vulnerable.

We analyse that many benign applications load malicious code at runtime. There are genuine reasons for this behaviour, which we will present in this section.
If an attacker is able to install an application that pretends to give the common framework, e.g., by persuasive the user to install an apparently benign application that internally uses the identical package name as the framework, then applications based on this framework will load the attacker's code as an alternative of the real framework. Without custom integrity checks, the applications will run the malicious code with their own permissions, and the attacker gains complete access to their internal data. The approach of loading a common framework was employed by a famous company developing web and multimedia software. They used it for their multi-platform application framework until June 2013[1], when they began to collection the framework with every Android application that uses it.
A few applications can be extended by add-ons that are installed as separate applications. Thus, malware can pretend to provide add-ons for legitimate applications, which those applications incorrectly load and execute.

A. Loading Additional Code

We now explain how the ability to load external code can lead to severe security issues. We present two types of attacks:

1) A malicious apps that is able to avoid detection by the Google Bouncer, so that it is made openly available on Google Play.
2) Code injection attacks against benign applications that use code-loading techniques, affecting millions of users [1].

IV. SECURITY THREAT REPORT

In this section, we explain security threat reported in 2013.

When users arrive at particular app download sites, users see a webpage that is carefully crafted to attract them to download and install a malicious app. For example, the user might be encouraged to install a fake update for products such as Opera or Skype. Or, in some cases, a fake antivirus scan is run, reports phony infections, and recommends the installation of a fake antivirus program. Once installed, the new app begins sending SMS messages. Many of these Trojans install with what Android calls the INSTALL_ PACKAGES permission. That means they can download and install additional malware code in the future.

For example, consider the malware-infected editions of Angry Birds Space we saw in April 2012. Again, available only through unofficial Android app markets, these Trojans play like the real game. But they also use a software trick known as the GingerBreak exploit to gain root access, install malicious code, and communicate with a remote website to download and install additional malware. This allows these Trojans to avoid detection and removal, while recruiting the device into a global botnet [5]. Through the use of a malicious Android app that harvests SMS messages in real time and in concert with a public engineering attack, attackers open a brief window of chance to take this a token and use it before you can end them [5]. Potentially unwanted applications (PUA) are Android apps that may not strictly qualify as malware, but may introduce security or other risks.

A. List of malicious Android apps

In January 2014 Kaspersky Lab had accumulated about 200,000 distinctive samples of mobile malware, up 34% from November 2013 – two months before over 148,000 samples had been recorded.
On January 30, 2014, the official Google Play market presented 1,103,104 applications. Alternative, unofficial stores have many more likely to be malicious. Kaspersky Lab has now logged 10 million unconvinced apps, as cybercriminals use also legitimate Android software to carry their malicious code.
In most cases malicious programs target the user’s financial information. This was the case, for example, with the mobile version of Carberp Trojan that originated in Russia. It steals user credentials as they are sent to a bank server. According to Kaspersky Lab experts, the majority of malicious Android applications are currently developed in Russia.

V. FRAMEWORK INVESTIGATION

In this work, we focus on the Assess of anti-malware products for Android. Specifically, we attempt to deduce the kind of signatures that these products use to detect malware and how resistant these signatures are against changes in the malware binaries.

In this work, we take several malicious app or malware samples and then passes through anti-malware tools. The detection results are then collected and used to make deductions about the detection strengths of these anti-malware tools.

A. What Does Anti-Malware Software Do?

The best antimalware software prevents and eliminates malware powerfully and efficiently. If you are the only one who uses your personal computer or laptop, smart phone, in most cases, antimalware software is all you need for reliable, vigorous internet security. It includes the same basic security features found in internet security suites without the extra elements, such as parental controls or password managers.

The best antimalware software defends your Smartphone's, tablets, computer from malware tricksters, such as viruses, Trojans, spyware, rootkits, keyloggers, pop-ups, scripts and adware.

The world's most popular anti-malware technology is Malwarebytes Anti Malware Mobile. Malwarebytes Anti-Malware Mobile protects your smartphone or tablet from malware, infected apps, and unauthorized surveillance.

Malwarebytes Anti-Malware Mobile

- Detects and eliminates malware, including spyware and Trojans.
- Scans your apps for malicious code or Potentially Unwanted Programs (PUPs).
- Stops unauthorized access to your personal data.
- Scans your Android device for security vulnerabilities.
- Identifies applications that are tracking your location.

Powerful anti-malware and anti-spyware technology protects your Android device. Detecting Trojans, spyware, and other Potentially Unwanted Programs (PUPs) before they can steal your identity, eavesdrop, or degrade your mobile experience.

Seven antimalware tools, which are listed in Table I. There are dozens free and paid antimalware offerings for Android from various entrenched anti-malware vendors as well as not so known developers. We selected the most popular products; in addition, we included Kaspersky which was not very popular but are well established vendors in the security industry.

TABLE I
EVALUATING ANTIMALWARE PRODUCTS

Product	Available
DroidDream Malware Cleaner	FREE
Kaspersky Internet Security	FREE
Fastscan Antivirus K-TEC Inc.	RS.123.56
Norton Security antivirus	FREE
Zoner Antivirus Free	FREE / $4.99
AntiVirus Security	FREE
Antivirus Mobile Security	$19.90

VI. CONCLUSION

Our analysis shows that the ability of Android apps to load additional code at runtime causes major security issues. We were able to show that an astonishingly large portion of existing applications is vulnerable to code injection due to download and access apps from unofficial sits. Additionally, we showed that attackers could use dynamic code loading to avoid detection in particular the Google Bouncer. In order to automatically detect such malwares, vulnerable or malicious functionalities, we had given the list of effective antimalware available in market.

REFERENCES

[1] Sebastian Poeplau, Yanick Fratantonio, Antonio Bianchi, Christopher Kruegel, Giovanni Vigna,"Execute This! Analyzing Unsafe and Malicious Dynamic Code Loading in Android Applications", NDSS '14, 23-26 February 2014, San Diego, CA, USA.

[2] Building a test environment for android anti-malware Tests.

[3] H. Lockheimer, "Android and security." [Online]. Available: http://googlemobile.blogspot.com/2012/02/android-and-security.html

[4] Rafael Fedler, Julian Schütte, Marcel Kulicke, "On the Effectiveness of Malware Protection on Android", *Fraunhofer AISEC April 2013*

[5] SOPHOS, "Security Threat Report 2013".

[6] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang,"Catch Me If You Can: Evaluating Android Anti-Malware against Transformation Attacks", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 1, JANUARY 2014

[7] http://www.kaspersky.com/about/news/virus/2014/Number-of-the-week-list-of-malicious-Android-apps-hits-10-million.

[8] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang," DroidChameleon: Evaluating Android Anti-malware against Transformation Attacks", ASIA CCS'13, May 8–10, 2013, Hangzhou, China.