Android Malware Behaviors for Android Platform Using Interactive Labs

Colin Szost, Kriti Sharma, Tae Oh, Willaim Stackpole and Richard P. Mislan

***Abstract*— Android is a fairly new operating system launched by Google in October 2008; it has been gaining market and growing popularity ever since. Alongside the growth of the operating system, malware for Android has increased tremendously. Currently, few strategies are available to identify and detect malware on this platform. Users are generally naïve and are fooled into downloading apps, which may be harmful or malicious. They do not have enough experience with either the platform or the apps to know what can be malicious and what is genuine. Thus, the goal of the authors is to create a series of interactive labs that instruct students with little experience with the Android platform or dynamic malware analysis techniques in the methods of dynamic malware analysis and dynamic malware analysis tools for the Android operating system.**

***Index Terms*— Android, Malware, Dynamic malware analysis, Botnets, Analysis techniques, Apps, Permissions**

I. Introduction

MOBILE DEVICES have become an integral part of the daily social fabric of our lives and they are the most popular target platforms for attacks. The past year has seen an over 2000% increase in unique malware variant attacks. Recent mobile malware studies have stated that mobile anti-virus tools are likely to catch less than 20% of these attacks. Furthermore, many devices do not have any mobile malware prevention tools installed. With always-connected capabilities, mobile devices have become the ultimate access point to a person's most private and personal information. Once a device has been maliciously compromised, personal data is accessible both locally (on the device) and over the network (in "the cloud"). To top it all, blurring the line between personal and professional is the critical issue relating to "Bring your own device" (BYOD), which impacts both the industry and government sectors.

As the use of mobile devices continues to rise in our personal and professional worlds, there is a growing need to educate future professionals in the topics of mobile security and vulnerability exploitation. Currently in the United States, there are more wireless mobile devices than there are people (331.6M/311.5 at 106%). As the ultimate human computer interface, the mobile device allows almost anyone to do almost anything anywhere. The mobile device transcends space and time through its multitude of communication, transaction, and entertainment tools.

As important and indispensable as these devices have become, security of mobile data and devices is an element that is still largely overlooked. There is a strong sense of urgency to build awareness and protection, and to prevent such mobile device threats. Currently, no academic institution focuses on the lack of security of these personal mobile devices as a threat to private and personal information. Though some programs may add the topic as a week of content into a single cyber security course, this approach is functionally inadequate in addressing the large volume of issues facing the mobile environment.

This paper discusses in detail four labs that have been developed by the author to familiarize the students with the Android platform and dynamic malware analysis techniques. Section II describes the history of the Android operating system. Sections III and IV describe the need for educating students in this field and the need for dynamic malware analysis. Section V describes the first lab, which introduces students to the Android emulator software. The remaining three labs, which introduce the students to dynamic malware analysis, are presented in sections VI, VII and VIII. They present various malware apps that send text messages without the permission of the user, apps used for advertising, and a simple mobile botnet. These labs teach live analysis techniques as well as the principles and methods used by the malware developers and how they trick users into running their malware. Section IX then suggests additional future work in this area and is followed by the conclusion.

II. History of Android Operating System

Google's Android operating system was released to the world in 2008 and is based on a modified Linux kernel, built on the ARM platform. Google claims that each day there are at least 1.3 million activations and the total number of Android devices exceeds 5 million making it the most widely used mobile operating system [1]. Estimates suggest that approximately 75% of the total smartphone devices shipped as of October 2012, were Android devices [2].

C. Szost is with the Rochester Institute of Technology, Rochester, NY-14623, USA (e-mail: crs1471@rit.edu).

K.Sharma is with the Rochester Institute of Technology, Rochester, NY-14623, USA. (e-mail: kxs1203@rit.edu).

Tae Oh is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: thoics@rit.edu).

Bill Stackpole is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: wrsics@rit.edu).

Rick Mislan is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: rpmics@rit.edu).

Android applications are written in Java and its lower level system utilities are written in C language. Its applications can also be written in C++ [3], [4]. However, Java is the preferred language. Google developed a custom "Dalvik virtual machine," which is the replacement for the Java virtual machine on the mobile devices running the Android operating system [5]. This Dalvik virtual machine is customized for use on mobile devices, which typically have limited resources. Thus, an application written in java is compiled into a "dex bytecode" in a file named *classes.dex*.

In an effort to keep the system secure, Android places all user applications into a sandbox when they execute. This allows applications to run independently of one another and not interfere with the resources and memory requirements of other applications. This is done using standard UNIX process separation techniques. As a result, each application is isolated from all others. Each application is assigned a User ID (UID) and a Group ID (GID) when they are installed. There are two ways in which users can install applications into their devices: by downloading from Google's play store or by direct download and installation – with an option to store the app on a memory card. While installing these applications the users are presented with a list of all the permissions that the app requires to be able to execute in the mobile device. The users then have two options: either accept all permissions and allow the app to install or disallow the app and choose to not install it. (No option is provided to the user to choose which permissions to allow or deny – it is an all or nothing selection.)

III. Need for Educating Students in This Field

As mentioned in Section I, malware has been growing tremendously alongside the growth of the Android OS. This calls for a strong need to evaluate the awareness of the users with what malware is and how harmful it can be to their personal and private information. The caveat here, however, is that users are naïve and do not have much of an understanding in this matter as to what apps are genuine and which are malicious. They do not know which permissions are genuinely required by an app and which are malicious and may be used only as a tactic to gain access to private and sensitive information. Most users, rather than reading through and trying to understand *what* permissions are being requested and *whether* it is appropriate for the app to be asking for those permissions, just go ahead and allow the app to access all the information it requested. This is the requirement to permit installation on their devices. Most malware developers make use of this naiveté on the user's part and develop malicious apps, hoping that users not know the difference, will not read through the list of permissions, and will just download and install the app.

This paper presents a concerted effort on the authors' part to educate students regarding the Android platform and to help them gain practical experience with it and its environment; so, when students go out in the world and are faced with such situations, they know what they are dealing with and have a strong conceptual background and basic understanding of what the system does and how it interacts with the apps.

In addition to learning about the Android operating system, the authors also want the students to better understand how apps interact with the underlying operating system, what connections they make, what permissions they request and what API calls they make. By answering those questions, the students can learn and are able to differentiate between genuine apps and malicious apps.

IV. Need for Dynamic Malware Analysis

Dynamic malware analysis is also known as behavior testing or live testing. This allows the user to see what the malware is doing in real-time. It allows the user to see and analyze the behavior of the malware as if it was installed and executing in the actual mobile device. During the dynamic malware analysis, the malware is installed in a virtual mobile environment using an Android emulator. The malware then tries to make the connections as it would if it were installed in a mobile device. However, since this device is virtual and is being monitored by the user, armed with sophisticated tools like tcpdump, users are able to see all aspects of malware behavior such as what processes are running, what network connections are made and what data is transmitted. Based on this behavior, the user can infer conclusions regarding the malware and see exactly what information it steals and how to protect against it.

V. Lab 1 - Android Emulator

The first lab is a brief introduction to Android virtual devices and the steps to set up the testing environment. The labs use a setup of two virtual devices: one innocent device installed by student used as a control, and one device infected with the malware.

The lab provides the student with detailed instructions for the setup for these devices and the environment. It also instructs the student on how to interact with the virtual phone through the adb shell. This gives the user command line access to the underlying Linux operating system on which the Android system runs. This is then used for gathering important information like what processes are running, network statistics, etc. by using the commands like ps and netstat. The ps command will display information about all the processes running on the device and will also include information regarding the name of the user to whom the processes belong to, process ID and the name of the process. This enables the student to see which app is running which processes and if it is normal behavior or if it is malicious. Netstat shows network connections the device has made or attempted to make and the addresses and ports that are in use. In the Android version of these commands are stripped down compared to the true Linux commands, but they can still offer valuable information.

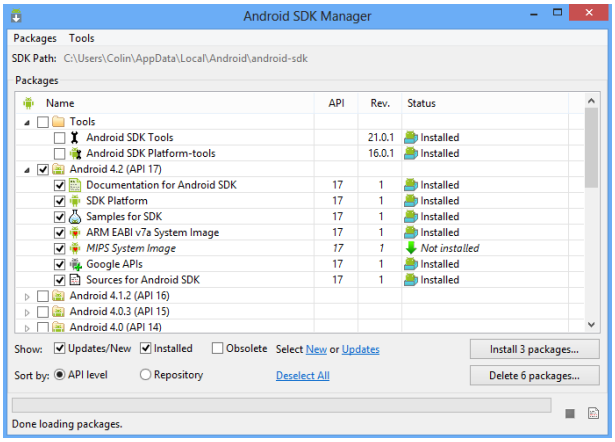

Fig. 1. Android SDK Manager Installation

Another tool introduced was tcpdump. This is the main tool used for analysis and can capture network traffic from a device, provided that the tool has "system" or "root" priviledges. While tcpdump isn't capable of detecting SMS data, it does show all the other connections that the phone makes. It will listen to any network traffic that the device sends or hears and saves it so that the user can look at it afterwards and analyze its content for malicious nature and behavior.

While this lab doesn't introduce any malicious applications, it gives the student a reusable lab environment to work with for the future labs as well as an understanding of how the tools and emulators work. This lab was designed for people who have little familiarity with both Android virtual devices and the Android platform in general.

VI. Lab 2 - SMS Malware

This is the first lab that introduces a malicious app. When picking a sample to use, there were two important characteristics, the first being a simple attack. Students may not be familiar with analyzing network traffic or even the Android platform in general, so it had to be something that wouldn't overwhelm an unfamiliar student. Additionally, a sample that could illustrate how much control a malicious application could have over the phone, and something that would be easily visible and would also be interesting for a student to observe.

A sample that met these criteria was the malware *DogWars – Beta*. This application did two important things. First, it contacted a website. It didn't send or request too much information and was not overly complex. It would give a simple introduction about why DNS would be helpful to malware creators rather than hard coding an IP address, and it would give them an example in what to possibly look for.

The other reason this malware was chosen is because it sends SMS messages without asking user permission. This is both very visible to the student, and more interesting than searching through packet captures. The goals for this lab were to show that a malware can send out mass messages without the user's interaction, knowledge or permission, and the student can see these effects. *DogWars – Beta* also demonstrated an SMS attack, which is one of the types of malware that students may encounter in the wild, and the more different types of malware a student is exposed to, the more they will know what is possible.

The following figure displays the tcp stream that was generated and contains important information regarding the malware network connections.

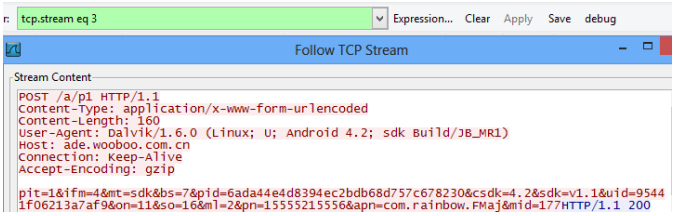

Fig. 2. TCP Stream

At the end of the lab, students are asked a series of questions that point them in the direction of some of the concepts they should be learning through the lab. This malware was rather simple, and the packet captures are easily analyzed.

While not overly complex, this lab gives a good introduction to using the environment, the tools, and the dynamic analysis process. It also demonstrates how easy it is for a malicious program to use a mobile device without the user's knowledge.

VII. Lab 3 - Show Me The Money

The third lab in the series tests a much more complex malware sample. The sample chosen was *LeNa.c,* the Android DFKBootKit variant. This application masquerades as a game. This worked well to show how a trojaned game can function almost completely like the real program while also performing malicious activity.

This application also is more complex than the malware used in lab two. The *LeNa* application sends and requests information in a much less obvious way over the internet to its controller's servers, including information sent in URL strings. The students would have to look for these strings in the packet captures as well as analyze what kind of information is being sent and requested, and determine the purpose of this communication.

The application is used to push advertisements to the user and replaces many of the *in-application* links with links that redirect to malicious websites.

More advanced analysis techniques are also introduced, such as some of Wireshark's advanced traffic analysis features and the student has to look in different places to find what the application is doing.

This sample also shows a different intent for a malware's purpose. While the first sample sent SMS messages as its malicious payload, *LeNa* is made for siphoning information from a device and pushing unwanted ads to the user. This illustrates the various purposes of a malicious program.

VIII. Lab 4 - Spam, It's What's for Spreading

The final lab in this series uses a sample of the Spam Soldier botnet. This is a simple mobile botnet with many useful features that make it ideal for educational purposes. The

main aspect that makes this a good malware sample for an introductory malware analysis course is that command and control functions are not encrypted. This allows students to be able to see and understand the commands being sent to the device without having to completely disassemble the malware.

During the analysis, students will be able to see the command and control messages in their packet captures, and how the application works when the commands are received.

For this lab, a custom command and control server must be set up so that the application accepts commands only from the lab server, and not a real malicious server. This allows students to look at the malware from both the infected device as well as the actual command and control server. This vantage point allows them to see how they interact and what the commands look like.

Another reason this malware sample was chosen was because it shows a different type of malware. Together, all three labs introduce a malicious application, including an SMS attack, a trojan, and a mobile botnet.

IX. Future Work

This paper is a stepping-stone to what the authors wish to be a full-fledged course, complete with many more detailed labs, which increase in complexity with regards to the malware categories and samples. Currently, these labs focus on SMS malware. In future labs, the authors will include malware samples from other malware categories, such as trojans, spyware, bots, and more. Future labs will also focus on helping students identify patterns and distinguishing features of each category of malware. This will enable students to get a deeper understanding of how each category of malware behaves and what they focus on as the end result of malicious behavior.

X. Conclusion

This series of four labs was designed to examine how malicious applications work, and how this malicious activity can be detected and analyzed. They were explicitly designed for those unfamiliar with both Android devices and malware analysis techniques. Currently, this is an incomplete series of dynamic malware analysis techniques labs as there are more advanced aspects of both malware and analysis to be introduced. For this, an advanced course is planned. As an introductory and supplementary set of labs, this initial series provides a broad overview of how common types of malware behave and operate.

The skills and tools learned in this series of labs can be transferred to a more advanced application, and can be used to test if a student is truly interested in this subject. The ideas and techniques taught in this course are also not strictly specific to Android malware, as they can also be used in examining traditional PC malware. Having this basic understanding and familiarity of mobile malware will be helpful to those looking for a deeper understanding of malware analysis topics, such as reverse engineering.

REFERENCES

[1] M. Burns, (2012 September 5). *Eric Schmidt: 'There are now 1.3 million android device activations per day'*[Online]. Available: http://techcrunch.com/2012/09/05/eric-schmidt-there-are-now-1-3-million-android-device-activations-per-day/

[2] IDC, (2012 November 1). *Android marks fourth anniversary since launch with 75% market share in third quarter, according to IDC* [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23771812

[3] Ohloh, (2012 November 29). *The android open source project* [Online]. Available: http://www.ohloh.net/p/android

[4] Google, (2012 November 1). *Android source code* [Online]. Available: http://source.android.com/source/downloading.html

[5] D. Borstein (2012 November 30). *Presentation of Dalvik VM Internals* [Online]. Available: http://sites.google.com/site/io/dalvik-vm-internals/2008-05-29-Presentation-Of-Dalvik-VM-Internals.pdf?attredirects=0