Intructional Perspective: A Course Module on Mobile Malware

Perron Johnson [1], Philip Harris[1], Keheira Henderson[1], Xiaohong Yuan [1], Li Yang [2]
[1]Computer Science Department
North Carolina A&T State University
1601 E Market St, Greensboro, NC 27411
[2]Computer Science and Engineering
University of Tennessee at Chattanooga
615 McCallie Ave, Chattanooga, TN 37403
USA
xhyuan@ ncat.edu, li-yang@utc.edu

ABSTRACT: *Due to the emerging popularity of mobile devices, topics on mobile computing and mobile security started to gain importance in university curriculum. Some universities created concentrated courses to teach these topics. Another approach is to develop course modules on these topics and integrate them into existing courses. This paper introduces a course module on mobile malware, which can be integrated into an existing computer security course. The course module includes a tutorial on mobile malware, and two hands-on labs. Our initial teaching experiences of this course module in an existing computer security course is discussed.*

Keywords: : Course Module, Mobile Malware, Trojan, AndroRAT

Received: 28 July 2014, Revised 3 August 2014, Accepted 10 August 2014

1. Introduction

Due to the popularity of consumer mobile devices and the rapid growth of mobile applications, many universities have introduced mobile application development courses in their computing curriculum, or are considering offering such courses to address the state-of-art technology (Alston 2012; Burd, Barros, Johnson, Kurkovsky, Rosenbloom & Tillman 2012; Jackson 2012).

Since mobile devices store a large amount of personal information about their users, such as contact and calendar information, location and call history, etc., and many organizations are adopting BYOD (Bring Your Own Device), mobile security has become a fast growing issue. According to the security firm Kaspersky Lab, nearly 100,000 new malicious programs for mobile devices were detected in 2013, which was more than two times the number detected in 2012 (Hilburn 2014). Therefore it is very important to educate mobile device users about mobile security.

There have been some courses or course material on mobile computing provided by organizations and universities. The SANS Institute offers a course "*SEC*575: *Mobile Device Security and Ethical Hacking*" which introduces topics such as mobile device threats, policies, and security models; mobile device architecture security and management; mobile code and application analysis, ethical hacking mobile networks, mobile phones, tablets, and applications, etc. (Wright 2014) The Stanford Center for Professional Development provides an online course "*Mobile Security*" which introduces the approaches that enterprises have taken to manage mobile device, threats to mobile devices, and threats posed by various mobile "*app stores*" (Boneh, Daswani & Mitchell). The Carnegie Melon University (CMU) offers a course "*Mobile Security*" which covers topics such as smartphone security; mobile Internet security; mobile location privacy; and ad hoc, mesh, and sensor network security, etc. (Tague 2013) Guo et al. are developing the Android Security Labware (Bhattacharya; Guo, Bhattacharya, Yang, Qian, Yang 2013) which includes a set of lab modules: Threats of Lost or Stolen Mobile Devices; Unauthorized Mobile Resource Access; Mobile Privacy Threats, Mobile Malware, secure Mobile App Development; Mobile Short Message Service (SMS) Security; and Mobile Phishing Threats. These lab modules can be integrated into different information security courses to introduce mobile security concepts.

The Department of Computer Science, North Carolina A&T State University adopted the approach of developing different course modules on mobile computing and mobile security, and integrate them into computer science courses. This paper describes a course module on mobile malware we developed. We focus on malware on the Android platform since most of the mobile malware have been on Android platform (Kelley 2014). This course module includes lecture presentation and hands-on labs. One of the hands-on labs is based on the "*Mobile Malware*" lab module from Android Security Labware (Bhattacharya; Guo et al. 2013).

The rest of the paper is organized as follows: Section 2 provides an overview of mobile malware in the Android environment. Section 3 describes a hands-on lab of exploring an Android Trojan, AndroRAT. Section 4 describes a hands-on lab of writing a simple mobile malware. Section 5 discusses our teaching experiences followed by conclusion in Section 6.

2. Mobile Malware Tutorial

The mobile malware tutorial introduces different types of android malware, real life android malware examples, how malware spreads, and how to prevent mobile malware. Since students will conduct hands-on labs on android malware, we also introduce hacking ethics so that the students understand the responsibility and acceptable conduct associated with the knowledge learned. The following introduces the concept of mobile malware and its governance.

2.1 Types and Malicious Logic of Android Malware

Mobile Device Data Stealer is a type of malware that steals personal information like contacts, browsing habits, GPS locations or SMS and phone calls. An example of this type of malware is called NickiBot (Jiang 2011a). NickiBot can perform GPS-based location monitoring, sound recording, email-based uploading, as well as call log collection. NickiBot will only execute certain function unless it is told by the server or SMS. Another example of mobile device stealers is the "*Find and Call*" malware on iOS and Android devices (Maslennikov 2012). When the "*Find and Call*" malware is started on your device, it will ask you to register your device online. Once registration is completed it will begin spreading to the contacts in the phone by sending SMS spam messages embedding an URL in the message. The malware will also upload the contact book to a remote server (Maslennikov 2012).

Rooting-Capable Malware is a type of malware that tries to take control of the phone by requesting root access. Once the malware has root access it will be very challenging to remove the malware. An example of this type of malware is DroidKungfu (Jiang 2011b) and its variants. The DroidKungfu can access arbitrary files in the phone and gain the capability to install or remove any packages which will result in malfunction of certain applications. Some minor variants of the DroidKungfu have the ability to change the user's default web page without users' notice. Another example of rooting capable malware is DroidDream (Bradley 2011) which is a high threat level malware that has rooted itself into many legitimate applications. It has the ability to exploit Android apps, root Android devices, and send sensitive information such as the International Mobile Station Equipment Identity (IMEI) to a remote server.

Premium Service Abuser is a type of malware that sends SMS messages or makes phone call to premium services that will charge the user. An example of this type of malware is Zsone (Strazzere 2011) which was found in China. Once the user runs the malicious app on his phone, the app will send an SMS message to subscribe the device owner to a premium-rate SMS

service. Txtnation (SMS Service) is an example of the premium-rate SMS service. It provides Bulk SMS alerts and reminders, SMS Billing for website monetization and premium rate SMS for content services.

2.2 How Mobile Malware Spreads

Mobile Malware can infect mobile devices by the following ways:

(1) Infect through Bluetooth devices. Malware can spread to other Bluetooth devices in surrounding areas.

(2) Users download from suspicious sites. Malware authors create fake websites to cause their malware being downloaded. Users are fooled into believing they are traversing a legitimate site and downloading a useful app.

(3) Repackage as Trojan. Malware authors locate and download popular apps, disassemble them, enclose malicious payloads, re-assemble and then submit the new apps. Malware authors have chosen a variety of apps for repackaging, including paid apps, popular game apps, powerful utility apps (including security updates) to further increase malware infection rate. Malware authors create meaningful names for their classes when the malicious code is injected into a non-malicious app to avoid detection on analysis (Jiang & Zhou 2013).

(4) Apps automatically download updates. This concept includes a function in the code which will download malicious commands at runtime. As a result; static scanning of host apps may fail to capture the malicious payloads. Furthermore this attack can break down into stealthy updates or user updates. The malware known as Plankton uses stealthy updates which do not need user permission. It directly obtains a JAR file from a remote server and only updates a few parts of the code. In the case of user updates, the user has to allow the app to download the new version of the app. The malware can infect a device to obtain access to the phone's functionality while the app is being updated (Jiang & Zhou 2013).

2.3 Google Play Store's Protection Against Mobile Malware

Google play store policy does not allow worms, virus, Trojan horses or malware to be uploaded to Google play store. However, the process of uploading apps to the play store does not completely prevent malware from being uploaded to the Google play store. Google's Bouncer software is a measure to prevent malware in Google play store (Hou 2012). Bouncer dynamically scans apps that are in a waiting queue to be put in the play store. It also continuously scans the apps in Google play store in the background. Based on reports from the Bouncer, the Google play store had a 40% drop in the number of malicious apps (Hou 2012). A major concern with bouncer is that if malware does not misbehave while being scanned bouncer will not flag it. Google does have a team in place to monitor its apps on the play store, however, once a malware manages to enter the Google play store it could have done enough damage before it was seen.

2.4 Malware Prevention and Detection Using Sandbox

Sandbox refers to an area of space that is separated from critical resources of a system, in which untested code is run. Static analysis tools as well as dynamic analysis tools can run within the sandbox.

Spreitzenbarth et al. (2013) proposed Mobile-Sandbox for detecting malicious source code statically and dynamically in an android environment. During its static analysis it retrieves the android manifest file and logs the permission the application needs. It then breaks down the delvik byte code files into smali and once completed it checks for dangerous function calls such as getSimCountryIso() which helps attackers contact the right premium services (Spreitzenbarth, Echtler, Hoffman, Friling & Schreck 2013). Mobile-Sandbox uses two tools which are TaintDroid (Enck, Gilbert, Chun, Cox, Jung, McDaniel & Sheth 2014) and droidbox (Droidbox 2014) to scan malware dynamically. The TaintDroid runs in the background of a device and monitors privacy sensitive information used in an app. TaintDroid will notify the user when sensitive information leaves the phone. The droidbox reports analysis results including information leaks via the network, file and SMS, circumvented permissions, cryptography operations performed using Android API, and etc.

Lee et al. proposed a mechanism to identify malware that uses packing and obfuscation to avoid anti-viruses. The API call sequence of the malware is converted into a call graph, which is reduced to a code graph. A code graph is used to uniquely identify the malware (Lee, Jeong, & Lee 2012).

Airmid is a prototype tool that automatically identifies and responds to mobile malware based on their network behavior. The software uses network sensors to detect malicious traffic by using traditional security tools (Nadji, Griffin, & Traynor 2011). Once the network sensor detects malicious traffic it alerts the device using authenticated channel. A program on the device identifies the executable code responsible and creates a plan of action to repair the device, which may include filtering the traffic at the device, sandboxing or removing the app, patching the device or restoring the device to its factory settings.

2.5 Best Practices for Mobile Device Users to Defend against Mobile Malware

Users can defend against mobile malware by following the guidance below (Conner 2011; SOPHOS):

(1) Always look at the permissions requested by the app and check to see if the permissions are feasible.

(2) Download from reliable sources.

(3) Install updates as soon as they are released. Many updates may include security patches or updates which can provide further defense against malware.

(4) Remove malware as quickly as possible so no damage or further damage can be done.

(5) Use password based authorization on the device. This will greatly reduce unauthorized users

(6) Install security software and anti-viruses such as Lookout, Norton and AVG.

(7) Do not root the device. Rooting may give malware control of the device.

(8) Encrypt data on the device. This prevents man-in-the -middle attacks.

3. Mobile Malware Hands-on Labs

3.1 Hands-on Lab 1: AndroRAT

AndroRAT is an open source remote administration tool (RAT) for Android devices which was developed as a proof-of-concept project (Goddard 2013; Lelli 2013). AndroRAT is compiled in the form of an Android APK, the application package file for Android based devices. It can be bound with a legitimate android APK to become a Trojan by using software such as APK binder. AndroRAT Trojan allows a hacker to remotely access the functionality of a mobile device infected with the Trojan and steal information from it. According to Symantec, AndroRAT or Android.Dandro listed in the malware repository was detected on January 29, 2013 (Neville 2013).

AndroRAT has two parts: AndroRAT Client (Figure 1) and AndroRAT Server (Figure 2). The AndroRAT Client runs on a mobile device as a simple application where the communication between the client and the server machine is initiated with a button on the app. If AndroRAT is installed as a Trojan, the communication begins when the application of the infected APK launches. This communication is unknown to the victim user. The AndroRAT client waits for a command from the AndroRAT server. When it receives the command, it executes the command by calling a method or event handler class. The AndroRAT server uses socket programming that provides the communication mechanism between the client and the server using TCP.

The AndroRAT Trojan allows a remote attacker to perform the following actions on the compromised device (Goddard 2013):

• **Contacts:** An attacker is able to view the victim's contact list, including information such as contact names, number, address and email.

• **Camera:** An attacker is able to activate the victim's front or rear facing camera to take pictures or record video.

• **SMS:** An attacker is able to view victim's SMS text and information about the text such as phone numbers, time, dates and text content. Text can also be forwarded to other devices by the attacker.

• **File Tree:** An attacker is able to access the file system of the victim's device. Files from the SD card or internal memory can be viewed and downloaded.

• **Web Browser:** An attacker is able open a web page on the victim's web browser.

• **Microphone:** An attacker is able to activate the microphone on the victim's device and record audio.

• **Phone:** An attacker is able to make phone calls from the server to any desired phone number through the victim's device.

• **GPS:** An attacker is able to utilize the victim's GPS to obtain their coordinates and real time location.

A hands-on lab was designed for students to explore the functions of AndroRAT. The students were given a menu on how to download AndroRAT client on an Android Emulator, download AndroRAT server, and run AndroRAT client and server. The students were also given a scenario of downloading files from Android Emulator through AndroRAT client. The students were asked to do the following:

1) Obtain contact information, SMS and call logs from the android device of the victim through the AndroRAT Server;

2) Create a second emulator (Emulator 2) to communicate with the first one (Emulator 1). Send a SMS to Emulator 2 from Emulator 1, and make a phone call from Emulator 1 to Emulator 2 through the AndroRAT Server;

3) Discuss all the functions AndroRAT Trojan can perform, and the potential risks AndroRAT Trojan can cause;

4) Create two scenarios describing how AndroRAT can be used for attacking or good purposes; and

5) Download and install several Anti-Virus applications to the Android Emulator. Run the Anti-Virus applications and compare the results.

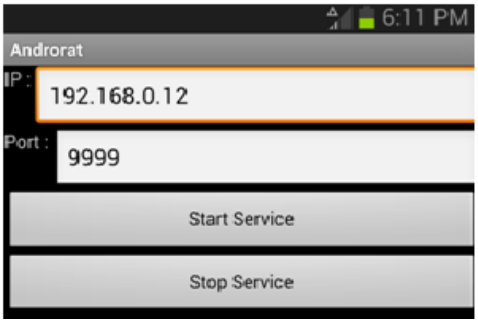

Figure 1. AndroRAT client running on a mobile device

Figure 2. AndroRAT server running on hacker machine

3.2 Hands-on Lab 2: Writing a Simple Mobile Malware

In this lab exercise, the students write an Android Trojan that sends text messages to other mobile devices. The malware makes users unaware of the malicious activities by deleting all the messaging history. This lab was adapted from the "*Mobile Malware*" lab module in Android Security Labware (Bhattacharya; Guo et al. 2013). It was a simplified version which is more suitable for students without intensive mobile programming experiences. The prerequisite of this lab is basic knowledge of Android programming.

In this lab students were asked to accomplish two tasks. The first task is to make a victim phone (Emulator 1) send text message to a pre-assigned phone number as the target, which can be an emulator's number (Emulator 2). The message was sent in a secret way so that phone user would not notice it. The second task is to delete any reply message to the text originally sent out by the victim phone so the phone user was not aware of any communication.

Students were given the following guidance regarding how to implement the assignment:

1) The program should include a class that will handle the incoming SMS messages called SMSReciever. The class must extend BroadcastReciever;

2) The program should have the following imports:

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;// handles phone actions

import android.os.Bundle;

import android.telephony.SmsMessage;//Handles phones SMS

3) The program must have permissions to receive and send text messages.

4) A reciever is needed which will alert the app if a text message has been received.

5) Once a message is received the app will parse the message and obtain its information.

6) In order to stop the phone from notifying the user that she has a text message, abortBroadcast() method will be used.

7) In the main activity of the app the student must use the SMSManger class which will allow access to SMS functions.

8) In order to get the phone number and other information from the phone, use the TelephonyManager class.

4. Teaching Experiences

This course module was taught in the course Computer System Security (COMP321) in the Spring 2014 semester. The class has twelve (12) students. This course module was taught in 3 class periods. Each class period is 75 minutes. In the first class, the students were given a presentation based on the mobile malware tutorial. The second class was conducted in a lab, where the students conducted the AndroRAT lab with the assistance of a student TA. AndroRAT software was already downloaded onto the lab computers to save lab class time. In the 3rd class, the students conducted the second hands-on lab in a lab. Most students were able to complete Lab 1 correctly. Most of the students needed more time after the class to complete the lab assignments. Some students didn't have any background on android programming. They were given reading materials on "*Introduction to mobile programming*" before the labs. These students had difficulties with completing Lab 2.

The main reason that some students could not complete the second lab is the lack of knowledge of mobile programming. Currently the Department of Computer Science at North Carolina A&T State University does not offer a mobile computing class. We are developing 12 course modules on mobile computing and mobile security, and integrating these course modules into existing courses. Mobile computing courses are integrated into CS1 computer programming course (COMP165) and a junior level programming course "*Programming Methodologies and Concepts*" (COMP365). Since this effort was started only recently, most students in the course of Computer System Security (COMP321) have not learned these course modules yet. Some students gained experience with mobile programming through internships or conducting research. After the course modules we developed are taught several times, the students in the COMP321 course will have the basic knowledge of mobile programming and will be able to complete Lab 2 without difficulty.

A survey was given to the students. However, the survey participation is low, and result is not useful here. The students were given two questions on mobile malware on their final exam. 90% of the students answered the questions correctly. We will continue teaching this course module, and assessing the effectiveness of this course module through survey and student performance on the lab assignments and on questions in exams.

5. Conclusion

Due to the popularity of mobile devices, the large amount of personal information stored on mobile devices, and the trend of BYOD, it is very important to educate mobile device users about mobile security. This paper describes a course module on mobile malware which can be integrated into a computer security class in a college computer science or IT curriculum. This course module includes a tutorial on mobile malware, and two hands-on labs. The mobile malware tutorial introduces different types of Android malware, real life Android malware examples, how malware spreads, how to prevent mobile malware, as well as hacking ethics. The hands-on labs include exploring the functionality of a mobile Trojan AndroRAT, and how to write simple Android malware software. This course module was taught in a Computer System Security course in Spring 2014. Our future work includes continuing improving and teaching this course module, assessing the effectiveness of this course module, as well as developing more course modules on mobile security that can be integrated in a computer science curriculum.

6. Acknowledgements

This work is partially supported by National Science Foundation (NSF) under the award HRD-1332504 and DUE -1241651. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

Citations and References

[1] Alston, P. (2012). Teaching Mobile Web Application Development: Challenges Faced And Lessons Learned, *In*: Proc. of the 13th Annual Conference on InformationTechnology Education (SIGITE '12), 239-244.

[2] Bhattacharya, P. SMART: Real World Relevant Security Labware for Mobile Threat Analysis and Protection Experience. Mobile Security Labware. Retrieved July, 21, from https://sites.google.com/site/mobilesecuritylabware/home

[3] Boneh, D., Daswani, N., Mitchell, J. (2014). XACS215 - Mobile Security. Stanford Center for Professional Development. Retrieved on July 8th, from http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=13070857

[4] Bradley, T. (2011). DroidDream Becomes Android Market Nightmare. PCWorld. Retrieved on July 9, 2014, from http://www.pcworld.com/article/221247/droiddream_becomes_android_market_nightmare.html

[5] Burd, B., Barros, J. P., Johnson, C., Kurkovsky, S., Rosenbloom, A., Tillman, N. (2012). Educating for Mobile Computing: Addressing the New Challenges, *In*: Proc. of the final reports on Innovation and technology in computer science education 2012 working groups (ITiCSE-WGR '12), 51-61.

[6] Conner, B. (2011). 10 Ways to Combat the Threat of Mobile Malware. TechRepublic. Retrieved July 9, 2014, from http://www.techrepublic.com/blog/10-things/10-ways-to-combat-the-threat-of-mobile-malware/

[7] Droidbox (2014). droidbox - Android Application Sandbox. Google. Retrieved on July 21, 2014, from http://code.google.com/p/droidbox/

[8] Enck, W., Gilbert, P., Chun, B. G., Cox, L. P., Jung, J., McDaniel, P., Sheth, A. N. (2014) TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, Communications of the ACM, 57(3) 99-106.

[9] Goddard, D. (2013). Androrat – Android Remote Access Tool. Vulnerability Research Team. Retrieved on July 21, from http://vrt-blog.snort.org/2013/07/androrat-android-remote-access-tool.html

[10] Guo, M., Bhattacharya, P., Yang, M., Qian, K., Yang, L. (2013). Learning Mobile Security with Android Security Labware, *In*: Proc. of the 44th ACM technical symposium on Computer science education (SIGCSE'13), 675-680.

[11] Hilburn, M. (2014). Cyber Thieves Increasingly Attack Mobile Devices. Voice of America. Retrieved on July 21, 2014, from http://www.voanews.com/content/cyber-thieves-increasing-attacks-on-mobile-devices/1860943.html

[12] Hou, O. (2012). A Look at Google Bouncer. Trend Micro. Retrieved on July 9, 2014 from http://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-google-bouncer/

[13] Jackson, S., Kurkovsky, S., Mustafaraj, E., Postner, L (2012). Panel: Mobile Application Development in Computing Curricula, *In*: Proc. of the 44th ACM technical symposium on Computer science education (SIGCSE'13), 107-108.

[14] Jiang, X. (2011a). Security Alert: New NickiBot Spyware Found in Alternative Android Markets. NC State University. Retrieved on April 2, 2014, from http://www.csc.ncsu.edu/faculty/jiang/NickiBot/

[15] Jiang, X. (2011b). Security Alert: New DroidKungFu Variants Found in Alternative Android Markets. NC State University. Retrieved on April 2, 2014, from http://www.csc.ncsu.edu/faculty/jiang/DroidKungFu2/

[16] Jiang, X., Zhou, Y. (2013). A survey of Android Malware. Android malware. New York, NY: Springer. 3-20.

[17] Kelly, G. (2014). Report: 97% Of Mobile Malware Is On Android. This Is The Easy Way You Stay Safe. Forbes. Retrieved on July 21, 2014, from http://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/

[18] Lee, J., Jeong, K., Lee, H. (2012). Detecting metamorphic malware using code graphs, *In*: Proc. of the 2010 ACM Symposium on Applied Computing (SAC'10), 1970-1977.

[19] Lelli, A. (2013). Remote Access Tool Takes Aim with Android APK Binder. Symantec. Retrieved on July 21, 2014, from

http://www.symantec.com/connect/blogs/remote-access-tool-takes-aim-android-apk-binder

[20] Maslennikov, D. (2012). Find and Call: Leak and Spam. *Kapersky Lab*. Retrieved on July 21, 2014, from https://securelist.com/blog/incidents/33544/find-and-call-leak-and-spam-57/

[21] Nadji, Y., Giffin, J., Traynor, P. (2011). Automated Remote Repair for Mobile Malware. *In: Proc. of the* 27th *Annual Computer Security Applications (ACSAC'11)*, 413-422.

[22] Neville, A. (2013). Android. Dandro. *Symantec*. Retrieved on July 21, 2014, from http://www.symantec.com/security_response/writeup.jsp?docid=2013-012916-2128-99

[23] SMS Service. SMS Service matters. Leading the pack across SMS services and solutions. *txt Nation*. Retrieved on July 14, 2014, from http://www.txtnation.com/sms_service/

[24] SOPHOS. When Malware Goes Mobile. *SOPHOS*. Retrieved July 9, 2014, from http://www.sophos.com/en-us/security-news-trends/security-trends/malware-goes-mobile/10-tips-to-prevent-mobile-malware.aspx

[25] Spreitzenbarth, M., Echtler, F., Hoffmann, J., Freiling, F., Schreck, T. (2013). Mobile-sandbox: having a deeper look into android applications. *In*: *Proc. of the 28*th *Annual ACM Symposium on Applied Computing (SAC'13)*, 1808-1805.

[26] Strazzere, T. (2011). Security Alert: Zsone Trojan found in Android Market. *Lookout*. Retrieved on July 21, 2014, from https://blog.lookout.com/blog/2011/05/11/security-alert-zsone-trojan-found-in-android-market/

[27] Tague, P. (2013). 14-829: Mobile Security. *Carnegie Mellon University*. Retrieved on July 21, 2014, from: http://wnss.sv.cmu.edu/courses/14829/f13/

[25] Wright, J. (2014). SEC575: Mobile Device Security and Ethical Hacking. *SANS Institute* Retrieved on July 8th, 2014, from http://www.sans.org/course/mobile-device-security-ethical-hacking