

1. Abstract

Android System is a mobile open-source operating system (OS), developed by Google, utilized by a large community of users from around the globe [1]. Due to its free and vast ecosystem, users with good intentions as well as criminals have taken advantage of the OS, unfortunately, implementing malicious attacks on many of Android's vulnerable applications [2]. Because of security risks and exposures facing Android OS, the primary concern has been exploring methods that enable Android to remain open-source and sustain a high level of security. As a result, this research investigates the recent **vulnerabilities** and security risks of Android System, in addition, utilizes one of vulnerabilities explored (CVE-2014-3500) to design and conduct an attack via application on Android mobile device. Essentially, this study will produce familiarity with how Android System security is approached and operated to keep the operating system secure for its users.

2. Introduction

Android System is an ever-growing open-source operating system (OS), made by Google, that is being applied through multiple devices such as tablets, mobile smart phones and a host of other manufacturers [1]. With this free environment application developers and users have the opportunity to learn how Android System functions and generate their own productive applications using Android Development Tools, with the benefit of reaching a large audience [2][3]. However, such an environment has not only attracted users with good intentions, but also criminals whose intentions are is to inflict harm or damage to users via Android's vulnerable applications.

Approaching the security risks and vulnerabilities in this research study highlight how Android System can continue as an open-source system and remain secure despite the threat of malicious attacks.

Investigating the Security Risks and Vulnerabilities of an Android System

Tatyana Matthews¹, Xiofeng Wang², Ph.D. ¹Elizabeth City State University, ²Indiana University

3. Objective

Investigate recent vulnerabilities of Android System that have led to

- its security breaches
- Design and conduct an attack using one of the vulnerabilities
- explored

Obtain familiarity with the approach to securing Android System

software for its users



5. Methodology

Research flow of methodology: 1. Develop abstract knowledge of how to perform a malicious attack 2. Research Android System's recent vulnerabilities using CVE tool and generate report 3. Identify vulnerability for malicious attack 3. Design attack and explore code 4. Conduct malicious attack using vulnerability

Activity Blogs Bookmarks F	Files Groups • More Elgg=p5o0rq93t17ir4v2trfb8mfcn4 Kice Brief descripti	2	1 <u>CVE-2015-3108</u> Adobe Flash Player Windows and before 18.0.0.144 on Windo mechanism via unsp 2 <u>CVE-2015-3107</u> Vse-after-free vulne An <u>CVE-ID</u> CVE-ID CVE-2013-4866
1224/product_id-1999 - <u>This means:</u> Any Ar can be launched throu can cause the Cordova different start page/ho	roid before 3.5.1 edetails.com/vulnerability-list/vendor_io 7/Google-Android.html udroid applications built with the Cordow gh a special intent URL. This specially- a-based framework application to common mepage than what the developer or user r other HTML content stored on the Andro 5	crafted URL ence with a intended.	<pre> (script> location.href =</pre>

6. Conclusion

Gained knowledge of approach to combating challenges rendered by vulnerabilities in Android System applications

Designed an attack using vulnerability CVE-2014-3500, Apache Cordova 3.5.0-0.2.1, and Android Studio (view Figure 2)

Investigated and reported recent vulnerabilities and security risks facing Android System

Assumed the role of "attacker" and learned how to perform malicious attacks using Cross-Site Scripting (XSS) and Cross-Site Request Forgery

(CSRF) methods

4. Tools

CVE (Common Vulnerabilities and Exposures) databases:

CVE Details The ultimate security vulnerability datasource

Android Studio (includes IDE, SDK tools, 5.0 emulator system) with Google APIs, Nexus 4 APK, Android 5.0 (Lollipop) Platform):



Fig. 1: Primary tools utilized

Bypass +Info 2015-06-09 2015-06-10 before 13.0.0.292 and 14.x through 18.x before 18.0.0.160 on Windows 18.0.0.143 on OS X and Android, Adobe AIR SDK before 18.0.0.144 or ows and before 18.0.0.143 on OS X do not properly restrict discovery of pecified vectors.

2015-06-09 2015-06-10 Exec Code rability in Adobe Flash Player before 13.0.0.292 and 14.x through 18.x adroid Common Vulnerabilities and Exposures Report The LIXIL Corporation My SATIS Genius Toilet application for Android has a hardcoded Bluetooth PIN, which allows physically proximate

attackers to trigger physical resource consumption (water or heat) or user discomfort -Link: http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html This means: "My Satis" is an Android application that within it has a

hard-coded Bluetooth PIN "0000". To be specific the Satis is a "smart" coilet; it's dominated by an Android application created by LIXIL corporation using Bluetooth as a communication device with the toilet.

"intent:mydata1#Intent;action=myaction1;type=text/plain;end";

g. 2: Research method

Implement correct and effective Java code into the malicious application that will change homepage and communicate with

Apache Cordova framework application

Figure 2

Perform future attacks on Android System using the latest Android

vulnerabilities reported



8. Acknowledgements

I wish to recognize Dr. Xiofeng Wang and his graduate students for their guidance, contributions, and help with completing this research. Also, I would like to recognize Dr. Lamara Warren for providing the opportunity to engage in a stimulating research experience as well as develop as a researcher through the Summer Research Opportunities in Computing (SROC) program.

[1] A. Todd. (2014, October 23). What is Android and what is an Android phone? [Online]. Available: https://recombu.com/mobile/article/what-is-android-and-what-is-anandroid-phone_M12615.html [2] S. Poeplau et al, "Execute This! Analyzing Unsafe and Malicious Dynamic Code Loading in Android Applications," presented at the Network and Distributed System Security Symposium, San Diego, CA, 2014.

[3] Download Android Studio and SDK Tools: Android Developers [Online]. Available: https://developer.android.com/sdk/index.html



7. Future Work

- Conduct successful attack using the vulnerability identified in

9. References