Evaluation of mobile security across various Android operating system versions

Selina Fahy
School of Design and Informatics
Abertay University
DUNDEE, DD1 1HG, UK

ABSTRACT

Summary

Android is the most popular operating system (OS) on mobile devices, making the platform a prominent target for malicious hackers. Through testing various vulnerable Android OS versions for different types of vulnerabilities, this research aims to determine the risks and possible mitigations. Labs and resources, such as Teaching Mobile Security, will guide the testing and provide supporting evidence for the conclusion.

Aim

The aim of this project is to test and evaluate the security of Android smartphones to understand the impact on users as well as to attempt to apply mitigations.

Method

Research will be carried out to find current mobile security issues and understand how pen testing a mobile device is undertaken. Specifically, the author intends to gain a thorough understanding of malware, such as ransomware and spyware, and its implementations on Android smartphones. By using static and dynamic testing techniques, it will be possible to cover substantial ground in terms of understanding and implementing malware and possible mitigations. Using static analysis, the code can be reviewed, letting any vulnerabilities be located and mitigations planned. While dynamic analysis allows the use of specific tools to speed up the process and allow for automatic testing and exploiting. Overall, these techniques will help locate security flaws, exploit them, document them, and attempt to mitigate them; this will allow for the levels of security to be determined and be evaluated in order to determine the effects against the users.

Results

The result of this project will be to test the effectiveness of the labs in terms of determining how secure Android smartphones are. It may not be possible to exploit every single vulnerability within the limited time frame and mitigate them; therefore, any significant issue countered will be theoretically measured and presented alongside the practical data collected.

Conclusion

This project will demonstrate the value of mobile security through testing and evaluating levels while showing the impact and risk presented to those who use them. Therefore, it is important that developers use and acknowledge new and improved methods of Android development and implementations.

Keywords

Mobile security, Android, malware, Kernel, Android OS.

1. INTRODUCTION

Android OS is the most popular operating system with a market share of about 73% (O'Dea, 2021), providing one of the largest available places where users can download apps and share information using applications such as web browsers, application stores, and so on. With mobile phones rising in popularity among businesses and everyday users, including some users owning more than just one, it is unsurprising that there is an increase in malicious hackers targeting the Android OS.

With constant evolution of the devices, mobile smartphones have now been developed to use many more applications. These services can include GPS, SMS, fitness applications, mobile banking, and so on. Also, with the wireless nature of mobile phones, this allows them to connect to networks and other mobile devices while outside. This can increase the number of threats for smartphone users. So, some of the top threats are mobile applications, web-based attacks, mobile networks, and mobile device security (Gontovnikas, 2021).
With such a rapid evolution of smartphones, it is to no surprise that there is a shortage in mobile security specialists as well as security learning materials (Guo et al., 2014).
Through this study the author will cover a few topics including the basics of Android sandboxing and its security model, testing Android smartphones, and using Android Debug Bridge (ADB) to interact with devices.

All of this will be conducted using Android Studio and Oracle VirtualBox. Android Studio is an application that allows for developers to create and run applications on an Android emulator. This is an area that is most important when considering the basics of Android, as this is where information will ultimately be collected and stored, and therefore an interesting place for malicious hackers. While VirtualBox is an application that runs Virtual Machines, this will allow the author to be able to run an Android disk image while attempting to pen test various areas such as volatile memory and the kernel.

1.2 Android Sandboxing and Testing

The author will investigate the use of sandboxes in order to securely test various applications and vulnerabilities within Android smartphones by offering an isolated environment. Sandboxes can have many different set-ups that allow for the environment to do different things. This can include causing a process to be stopped at a particular section in order to be investigated, or instead a process can be aborted in order to prevent any harm to the system or host, or more interestingly, it can monitor and record system activities of a running application in order to gather information. Furthermore, there are many types of sandboxes such as user-space sandboxes and kernel-space sandboxes. These allow testing of applications with malicious intents to be located and tested at different levels of the OS.

The author plans to use sandboxes in order to test and exploit vulnerabilities in Android smartphones.

1.3 Android Debug Bridge (ADB)

The Android debug bridge is a command-line tool that will allow the author to be able to communicate with the Android emulators. Through this the author will be able to debug apps and run various commands that will allow for data and so on to be observed.

2. BACKGROUND

After being bought by Google in 2005, the public release of the Android mobile system was in 2008. However, it was in 2006 that the first mobile applications that required to be signed were released. This security feature was then followed by the release of other security features like address space randomization, sandboxing, biometric identification standards, real-time kernel protection, etc. (Snyder, 2019).
The most common section of Android smartphones that is compromised are applications. Applications, or apps, consist of several sections that equally function the app as well as lead to potential malicious attacks. One such section is the "activities". All applications have at least one activity called the 'main' activity, which gives form to the application and provides the user interface. The activities have the ability to change state based on user input or mobile actions; for example, an application may become 'paused' if the mobile phone receives a phone call.

Another section is "services", which runs in the background to start and run services behind the user interface, such as checking for updates every 10 minutes.

The final sections that are largely used in the creation and usage of an app are broadcast receivers and permissions. The broadcast receiver is the main method of communication between apps and between the app and the smartphone. Apps can state interest in receiving data and the OS will offer any available data to the app and is also used to send data at the app's discretion to other apps or the OS.
Certain permissions are required to be requested in order to completely use some API functions due to privacy concerns. For example, messaging apps require the permission to access contacts as it will be handling potentially sensitive data about you and your contacts (Hamandi et al., 2021).

A common attack against Android smartphones in order to obtain information from applications, and/or data about the OS is through malware attacks. These attacks can include viruses, worms, trojan horses, and many more. These programs are often sent through the web or pre-programmed into malicious apps that trigger upon download or run time and are usually built with the intent to steal data, harm the device, control the device, etc. These malware attacks can be broken down, understood, and observed through the use of a couple of techniques.

One technique that allows for the analysis of malware is static analysis. This allows the author to look at malware that has been specifically made for Android software without needing to run it. This is usually completed through code review, string search, obfuscation checks, and so on.

Another technique is dynamic analysis, this allows the author to be able to run malware with the intentions of documenting its actual functionality in a safe environment. Both mentioned techniques will occur in an isolated environment to protect the host machine when running the malware, this is called sandboxing.

There are several ways that security professionals can test and protect their smartphones from potential malicious hackers, such as manual testing, automation testing using tools, code review, web service testing, and application testing. Some of the most popular tools that are used by professional mobile testers are OWASP ZED attack proxy, Android Debug Bridge, Smartphone Dumb apps, and so on.

3. METHOD

The main aim of this project is to test various Android OS versions for vulnerabilities. Using different labs, security levels of various Android OS versions will be tested.
This report aims to capture the process of exploiting vulnerabilities, explain the technique behind it, and discuss and implement mitigation methods.
In order to achieve this, the following objectives must be met:

- Research about Android operating system, past and current issues, and features, and to better understand the method and tools used in order to test for vulnerabilities.
- Find and use Android images as well as construct images to pen test them in search for vulnerabilities and prepare the use of tools that will be used.
- Reporting and evaluation – reporting all findings regarding located vulnerabilities; evaluating them in terms of how easily they are exploited and how they affect day-to-day users and considering mitigation techniques to better secure the users' data.

3.1 Research

Firstly, before looking at the practical part of this dissertation, it is important that the author gathers more detailed information about Android, the OS, vulnerabilities and how to exploit them, and potential mitigations. Relevant books, papers, journals, etc. will be consulted with the intentions of building solid footwork for the practical aspect of the project. Key areas of the authors in-depth research are understanding the basics of Android, current and common vulnerabilities in Android smartphone devices; exploiting those vulnerabilities and understanding and implementing possible mitigations. Tools will be investigated and prepared for their ability to support the practical aspect of the project.
Using Teaching Android Mobile Security (Lalande et al., 2019) and OWASP Mobile Security Testing Guide (Mueller et al., 2021) it was possible to do thorough research about Android smartphones and possible vulnerabilities. Teaching Android Mobile Security (Lalande et al., 2019) covered the basics in terms of providing the knowledge to understand the vulnerabilities that can be found in Android smartphones and how to potentially exploit them.

3.1.1 Common and high-risk vulnerabilities

3.1.1.1 App permissions issues

This covers the area of application design. Developers give and ask for permissions for their application from the user in order to maximize functionality of said app. However, granting permission comes with the risk that too little or too much data is being handled by the application and either crashing or potentially having a significant amount of information stolen/leaked.

3.1.1.2 Insecure transmission of sensitive data

This will cover the concept of 'transport security' where data is transported using SSL/TLS etc. However, it may be common that developers do not successfully implement these measures which lead to weak encryptions, ignoring security warnings or certification warnings, etc. This is a simple vulnerability that is easily exploited.

3.1.1.3 Kernel

This area will cover the kernel of Android smartphones. It is the best choice of attack in order to gain full control of an Android smartphone. Understanding the extraction methods and being able to implement new kernel code. In order to understand this area, the author will need to further research the Android kernel and its role.

3.1.1.4 Radio interface layer

This area handles cellular communication by providing an interface to the cellular modem in order to work with the mobile network to provide mobile services. Through research the author will be able to determine the level of security that has been provided throughout the various versions as well as evaluate the impact against users.

3.1.1.5 User space software

This area covers stack overflow, heap overflow, and the basics of other memory corruption forms. The research will encompass the extent of memory corruption, and the effects that it would have on the user, the malicious hacker, as well as any applications on the smartphone.

3.2 Implementation

After thorough research and learning of mobile security, vulnerabilities, exploitations, and potential mitigations, the author plans to put this knowledge forward into testing virtual emulators of various Android OS versions. The testing will be handled through Android Studio, in a Linux environment, and VirtualBox. The author intends to investigate malicious malware that targets apps, web-browser data, and the like, including spyware, trojans, etc.

The emulators will contain fake information about the 'user' of the Android smartphone.
The tests will initially consist of loading an emulator and statically dissecting various malware that will be tested. Then proceeding to sandbox the emulator in order to dynamically assess the malware and the effects it had by checking if any of the 'data' had been stolen.

With the large number of possible Android mobile security vulnerabilities, it may not be possible to test all vulnerabilities; therefore, there will be a method followed that will allow the author to select which vulnerabilities and exploits that are the most useful.

3.3 Evaluation

Once the appropriate labs and exploits have been selected, the author will move forward into testing the Android smartphone emulators. Each version will be deployed either using Android Studio and/or by using Oracle VirtualBox to create an emulator for the smartphone. As various versions are being tested the security levels are presumed to vary; however, by comparing the levels together it is possible for the author to quantify the risks that each version holds to a user, therefore allowing the author to evaluate the security among them.

Following the exploiting of possible vulnerabilities in Android smartphones, the author will look at implementing mitigations. These implementations can consist of secure mobile coding, looking at creating 'tamper-resistant' applications and so on.

4. SUMMARY

In summary, this project will evaluate the various levels of security that can be found across Android OS versions. With smartphones being ever present and Android continually growing, it is important that secure applications and OS are considered as malicious users put not only a company at risk but also the data of other users. Ultimately, this project aims to provide background and practical understanding to the dangers that insecure mobile devices hold to users and offer possible mitigations.

5. REFERENCES

Drake, J., 2014. *Android hacker's handbook. Indianapolis: Wiley.*

Gontovnikas, M., 2021. The 9 Most Common Security Threats to Mobile Devices in 2021. [online] Auth0 - Blog. Available at: https://auth0.com/blog/the-9-most-common-security-threats-to-mobile-devices-in-2021/ [Accessed 8 October 2021].

Guo, M., Yang, M., Qian, K., Bhattacharya, P. and Yang, L., 2014. Learning mobile security with Android security labware | Proceeding of the 44th ACM technical symposium on Computer science education. [online] Dl.acm.org. Available at: https://dl.acm.org/doi/pdf/10.1145/2445196.2445394 [Accessed 9 October 2021].

Hamandi, K., Chehab, A., Elhajj, I. and Kayssi, A., 2021. *Android SMS Malware: Vulnerability and Mitigation*. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6550526 [Accessed 15 October 2021].

Lalande, J., Chaoui, H., Mazurczyk, W. and Berthome, P., 2019. Teaching Android Mobile Security | Proceedings of the 50th ACM Technical Symposium on Computer Science Education. [online] Dl.acm.org. Available at: https://dl.acm.org/doi/pdf/10.1145/3287324.3287406 [Accessed 8 October 2021].

Mueller, B., Schleier, S., Willemsen, J. and Holguera, C., 2021. MSTG Mobile Security Testing Guide. OWASP.

O'Dea, S., 2021. *Mobile OS market share 2021 | Statista*. [online] Statista. Available at: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/ [Accessed 8 October 2021].

Pocatilu, P., 2011. Android Application Security. [online] Revistaie.ase.ro. Available at: https://www.revistaie.ase.ro/content/59/14%20-%20Pocatilu.pdf [Accessed 9 October 2021].

Snyder, J., 2019. The Evolution of Mobile Security Solutions. [online] Samsung Business Insights. Available at: https://insights.samsung.com/2019/02/21/the-evolution-of-mobile-security-solutions/ [Accessed 8 October 2021].