Mobile Security: A Look into Android

Aaron Mos, Md Minhaz Chowdhury
Department of Computer Science
East Stroudsburg University of Pennsylvania
East Stroudsburg, PA, USA
Email: amos@live.esu.edu, mchowdhur1@esu.edu

***Abstract*—Mobile devices are used almost every day by a large population of the world. If your device is not running Apple's operating system, it is most likely running Google's Android operating system. We see Android OS in phones, tablets, and e-readers. Since Android OS is so popular, it comes with a lot of security issues, being open source comes with a lot of threats from malicious users that have access to the software. Due to this, Android users must take extra precautions when it comes to protecting their devices. This paper will discuss what these threats to Android devices are and how someone would go about protecting their devices.**

Keywords – APK (Android Application Package), Malware, Android, Malware Families, Mobile Security

I. INTRODUCTION

The usage of Android devices can be seen heavily in the phone market today. Phones like the Samsung Galaxy s10, the Samsung Galaxy Note 10 and the Google Pixel 3 all support the Android 10 update. These devices are fast, have great camera quality, have long batteries, and to be honest, are expensive. It is known today that with the power that comes in our phones, they are not just cheap equipment that can be replaced easily if something were to happen to them, which is why it is a necessity to keep our devices safe. Commercials are produced showing phone users that will keep using a phone despite dropping the phone numerous times, doing a lot of damage to the phone by adding cracks, blemishes, or other screen problems because of how expensive it can be to upgrade a phone. Yet, many do not understand the damages that already are subject to their phones every day when they do not take the time to keep track of malicious activity on the device.

Google's Android operating system is open source, meaning that anyone can have access to the files of anything that is created on the Android operating system, due to it having to also be open source. For example, games are a fun way for people to interact with their phones, but they are also a perfect way for a hacker to gain access of the phone. Attackers take the APKs of popular android app store games, add malicious behaviors or add multiple different groups of old malware code lines, then send them back to the app store or try to get people to install their malicious version of the app through advertisements [8]. Once these apps are downloaded on a phone, the phone will disassemble the packages letting the malicious packets access the rest of the phone's files. After gaining access to the files on your phone the malicious malware could do anything. For example, KMin is a malware family that changes your wallpaper or Plankton another malware family that will uninstall the current mobile security app on your phone. [4]

Mobile security is especially important today with how much time people are using their phones daily. For instance, with people's hectic schedules many do a substantial amount of mobile banking. Either through accessing their bank accounts through their banks own application or using third party apps like Venmo that allow the transfer of funds to friends on the app. If a user slips up and downloads a malicious app from an ad, a hacker can then gain access to their banking information through their phone's app, which leads to the potential for a lot of damage.

Children are also a massive concern when it comes to mobile security. Adolescent children as young as two or three and onward are seen with their parents' phones or their own mobile devices, many of which have no idea what they are doing but just click buttons while loading different YouTube videos and games. Even though, parents may think that their children could not do anything bad with their credit card information, or that their children are safe with a parental control. Is simply is not the case, their children could download malicious software onto the devices leaking all information or even giving full access of the parent's phone through backdoors placed by the hacker.

This paper will further discuss the topics of, what is android malware and how is it created in section one. What are malware families in and a closer look at "DroidKungFu" a particularly known dangerous malware in section two. Known threats and weaknesses that are presented on your Android system in section three. What can be done to protect mobile devices, is in section four. Lastly in section five will be a conclusion of the findings.

II. BACKGROUND

This section discusses about the android malware and how is it created.

1. Android Malware

Android malware is a piece of code with the intent to cause harm or steal information of the victim using an Android OS device. Since the Android OS is an Open Source operating system anyone has access to APKs of any app of the OS. This causes the Android OS to be quite unsafe, with the click of one download button you could download something that gives out root access of your phone, giving complete access to all your documents and photos. Mobile devices comparably to computers come in contact with a variety of different malware. Just to name a few, Spyware and Adware gather information about the victim and send it to third party sites where it is sold to target specific ads or gather

978-1-7281-5317-9/20/$31.00 ©2020 IEEE

information about a person or their computer. Worms, Trojans, and Viruses also affect Android users by getting downloaded through seemingly safe applications that wait out a specific condition to release its malicious code. [6]

Due to the open source requirement of all applications on the Android operating system, it is [1]easy for a hacker to gain access to the necessary vessel for their malicious code. For example, one hacker was able to gain access to the Angry Birds APK where they then embedded malicious code that would secretly send SMS messages that would charge the users 15 GBP per text, with over one thousand estimated to be affected by the attack. [2] Once installed a malware can accumulate permissions until it gains root access installing other malwares like spyware to monitor your devices.

2. Malware families

Malware families are a way in which different Android malware strains are classified based on common malicious behaviors. A malware family is forged when a new malware is created over an existing malwares code. The process of malware detection is done by analyzing patterns through signatures. [8] However, this method of detection has become less adequate due to malicious coders using older malware. Through obfuscation, the hacker changes some of the code while keeping the malicious bit intact. Thus, changing the bits of the signature while having the same behaviors of the old code. [6] This is specifically important for security analysts because if they have the same malicious old malware code then the methods of fixing the problem could be similarly to those of its families' and can be used as a guideline. Recently, there has been a drop in unique malware families but an increase in variety of malware. Hence, malicious hackers are enhancing previously developed malware to be unrecognized easier. [5]

This leads to malware that is not increasingly difficult to get rid of when it is found but malware that is not only harder to detect but also harder to classify into different families. [3] Dendroid is just one approach that has been developed to try and combat these problems, through analyst the creators propose that through a text mining approach they can classify automatically different smartphone malware into families based on code structures. [3] This would help with getting rid of the problem before it is able to wreak havoc on the victim's phone.

3: Droid Kung Fu

Droid Kung Fu became particularly popular in 2011 when it was first discovered. Not only because of its effectiveness in staying hidden but also its incredible use of encryption on RATC (or Rage Against the Cage) resource files to hide the malicious payload. [10] Droid Kung Fu uses AES-encryption to ensure that the payload is not discovered upon installation.[1] Once decrypted at runtime, the malware executes its exploits dynamically waiting for the proper time creating more obfuscation. This approach was so effective that when the first version of Droid Kung Fu was found no anti-virus software was able to detect it. [10] Droid Kung Fu uses its exploits to gain an increasing amount of privilege until it has gained root access, it also creates a backdoor for which it will receive commands. Through this backdoor the hacker has complete control of running programs and searching the web in the background undetected. Droid Kung Fu can also hide itself in many different disguises, it has been seen as a google search module, a google update, and puzzle games.[1][10].

Over the span of four months, six different families of Droid Kung Fu had been discovered. For each new family came new protection methods gaining more encryption on the backdoor and communication between the hacker and your phone. For instance, Allowing the malware to have stealthier installation of payloads through apps and pictures. This is just one example of how malwares are evolving to become stealthier and more lethal.

III. THEATS AND WEAKNESSES

In this section, known threats and weaknesses on Android system, are presented.

Threat 1. Premium Rate Calls and SMS

This threat charges the victims phone bill by sending premium calls and SMS. The main goal of this threat is to cause a financial burden on the victim, the hacker does not get any money themselves. An example of a premium rate malware was Fakeplayer, it would charge the victims by sending the message "798657" to multiple premium numbers. [9]

Threat 2. Search Engine Optimization

This threat causes the victims phones' search engine to artificially search for certain domains and raise the search ranks of specific websites by creating fraudulent clicks. [9] This leads to those specific websites having a greater lucky hood of being chosen when that topic is searched for by others, even if they do not have the best information out of all the other websites.

Threat 3. Botnets

This threat causes a group of Android devices to be under the control of one botmaster to which through a call and command center they give the bots commands to do a variety of tasks. For instance, these bots can be set up to wait for the user to launch a bank application for when the user is going to send a transaction through their banking application on their phone so they can intercept the information for their own use. [7]

Threat 4. Ransomware

This threat causes the loss of access of the data within a device until the ransom amount that is asked to unlock the device is paid for. FakeDefender.B [7] is an example of a ransomware that requires the user to pay the ransom in order to unlock the phone. Ransomware is quite tricky because there is no guarantee that when you pay the ransom that the device will come back and be working as normal. Once the ransom is paid the malware is still on the device and the hacker could just make the user have to pay again for their data if the malware is not removed properly.

IV. COUNTER AND SAFETY MEASURES

This section presents the ways that can be followed to protect mobile devices.

Appraoch 1. Static Approach

The Static Approach to malware detection is useful when checking for malicious behaviors of a file without executing the condition to trigger the malicious code. It does this by disassembling and analyzing the source code, described in the following section. [7] [11]

1. Signature Based Approach

In this approach, code is examined for a patterned segment called the signature. Static properties include hash signatures, packer signatures, header details, embedded resources and meta data. [13] If the signature matches one of an already known malware family, it will be discovered. This is a quick process that can help security analysts determine whether they should talk a closer look at the certain strain, to conduct further dynamic testing. [13] However, its' flaw comes that it is easily bypassed. Using obfuscation hackers can change the signature of the malware so it will not be detected by the signature approach method due to the new signature not being in the database. The signature-based approach also cannot help fighting against Zero-day attacks since it cannot detect polymorphic malware. [12] As the signature of the malware family will be unknow. Yet, the signature-based approach is quite effective in detecting already known malware family of files.

2. *Permission Based Analysis*

In this approach, the permissions of an app are checked to see if it is asking for more permissions that it really requires and to see if it is taking permissions for malicious reasons. Every app must contain what permissions are required by it in the AndroidManifest.xml file. [7] However, many hackers often ask for more permissions then the app really needs which can be a clue that the file is malicious. On the Android O.S, permissions play a monumental role in the protection of your devices. Unless otherwise changed by default apps have no permissions to access the user's data, the user just grants the access to the app. [13] The flaw with the permission-based analysis is that it only takes into account the permission manifest file. If the permissions are similar to a benign application a malicious one still could get through. Not to mention, that this is a static approach so it would not be able to detect malware that dynamically loads its payload like DroidKungFu. It would be appropriate to do a second run with another method after this one.

Approach 2. Dynamic Approach

The Dynamic approach also known as the behavior-based analysis method examines the app during execution. The use of debuggers, decompilers, and disassemblers assist in breaking down and reversing the code. Then, execution of the code happens for a short period of time to see if any malicious activities happen. Although slower than the static approach the dynamic approach is better equipped to handle obfuscation and encrypted payloads being able to detect polymorphic and metamorphic malwares. [13] However, its drawback is that, it is highly resource intensive. Also, if a malware has behaviors that are not exactly what the test is looking for it may be over looked.

1) Anomaly Based Detection

In this detection method, applications are monitored to check the behavior of the code. They are determined to be either benign or malware through the purposed checking of log files, battery level, CPU usage and other overall characteristics of the application during execution. Applications are monitored using machine learning algorithms to detect patterns in malicious applications. [13]

In one purposed method of anomaly-based detection named AntiMalDroid, code is taken during execution and through behavioral examination is determined and categories based on whether its' behaviors were benign or malware. [7] Next, the method takes each benign and malware and puts the code in a learning algorithm to create a signature for the behavior of the code. It then stores and compares the signature that is generated to signatures that are already stored in the database to consider if it is truly benign or malware. The resulting will either be benign already stored in the database, malware that is already stored in the database, or a new signature behavior that will be able to assist in later detection. This proposed model creates a dynamically sized database that will continue to grow when new behaviors are created. However, this method takes a lot of resources and time.

2) Data and control flow analysis

In this approach, malware is discovered by movement of sensitive data required by or moving from the application. TaintDroid proposed by Enck et al. [7] is designed to place markers on data that are requested from sensitive sources. For instance, the devices GPS, microphone, camera, and other sensitive locations. It can then identify data leakage from third party applications.

3) Emulation based analysis

In this approach, the security analyst uses a virtual box or a sandbox to guarantee that they maintain a higher permission level than the application. Instead of executing the malware on the same computer they open a new environment for the malware and antimalware to act upon each other. [13] Thus, the security analyst will see everything that is happening to the OS, the files and see if any permission escalation attacks are happening.

For instance, DroidScope is an Android emulation detection system based on Virtual Machine Introspection and built upon QEMU. [7] It allows a virtual space for the malware and antimalware to react to each other. All while DroidScope is monitoring the operating system to watch for changes in a safe space. Droid Kung Fu that was talked about earlier in the paper was discovered by this method. [13]

Another proposed emulation detection method is named "Android Application Sandbox (AASandbox) which uses both static and dynamic analysis. It begins by extracting a .dex file which it then preforms a static analysis to. [13] After it creates a private environment for dynamic testing on the executing program using the Money tool. This method cannot detect new malware types because of the use of static analysis when selecting what parts of the application to look at. It does not consider everything besides the .dex file.

4) Permissions management

When an app is installed it will have no permissions unless given. [7] However, without the installation of a third-party application the only available options for a user when asked about permissions for an application are allow the application permission or not allow. To which usually follows with the user just getting prompted a second time for access.[9]

For instance, a game you play on your phone may ask you for internet access, to which you feel is okay to accept. However, then it asks for camera, contact list, SMS, and call access, to which you do not see why a game should have access to all of these things, if they are not even required for the game. Yet, you cannot accept one without all of them or you may have to accept them all to access the game without permission acceptance notices blocking your screen. Using these third-party applications will give you access to changing the permissions you want to give for an application along with automatically keeping logs of every permission required by every application on your phone. [9]

5) Device locking

While device locking does not do anything for malware being installed onto your device or getting rid of already installed malware it does protect your device from being tampered with while you are not around it. There are options for fingerprint, PIN, and pattern locks. They each have their own issues of being socially engineered by looking at the finger prints for the PIN and pattern lock options. There are also applications on the google play store that can give your device more protection than the default provided login protections. As well as giving access to your fingerprints to your device. However, having a protection method on your phone whether PIN, pattern, fingerprint or an application from the google store will give your phone more protection from social engineering attackers.

6) Anti-virus

Although anti-virus should not be the only protection that you should be practicing on your Android device it is always a good idea to have an anti-virus as a failsafe. [9] Android devices now have the same options for anti-virus software as Windows users do. Companies like AVG, Lookout, McAfee, Norton all provide anti-virus solutions to have on your mobile device. As you would expect, the applications function similarly to the computer versions. There are options to scan your device, to which a report of found or not found malware will be displayed.

7) Installing from trusted packages

Being on the Android OS comes with the ability to install third party or non-Google Play Store applications without the require of being jail broken and loosing warrants. With many malwares coming for third party Chinese stores a possible protection method can be the verification of the contents of the APK files of your downloads. For instance, a new application has been created called APK Inspector, which will scan your asset, certifications, and resources of the incoming files APK to ensure it is verified before being sent to you. [9]

The above ways to protect mobile devices are promising and applicable to various cyber security, data mining and machine learning fields. [14-25]

V. CONCLUSION

Our phones are with us 24/7, we have them with us everywhere from the mall to our bathrooms. We keep our entire lives on these devices, yet we do not treat them like they hold our entire livelihoods on them. The question then becomes why do we not secure them like we should for the sensitivity of the information on them. This paper showcases that Android malware is not only much less defended than on computers or apple devices, but they are also evolving to become more deadly and more difficult to detect. Although as stated in the paper there is a lot of work being done to create a more efficient method of detection when it comes to these malwares. With the nature of the Android operating system being open source there is always going to be a constant battle between the evolution of malware and the evolution of the detection methods.

Work is being done to protect our devices and proactively combat the evolution of malware through different methods of detection like behavioral analysis and emulation analysis. However, the overall best protection methods for your device must be implemented by the user. Permissions are a huge key in the protection of your device, if applications are asking for too much permissions or something that you believe they should not have do not grant them permission. Always keep your devices safe and password protected, social engineering attacks can leave your data vulnerable even before the software.

REFERENCES

[1] H. N. D. C. Fan Wu, "An Overview of Mobile Malware and Solutions," Journal of Computer and Communications, vol. 2, no. 12, p. 9, 2014.

[2] N. B. A. R. S. L. C. Kimberly Tam. Ali Feizollah, "The Evolution of Android Malware and Android Analysis Techniques," ACM Computing Surveys, vol. 49, no. 4, p. 41, 2017.

[3] J. E. T. P. P.-L. B. A. Guillermo Suarez-Tangil, "Dendroid: A Text Mining Approach to Analyzing andClassifying Code Structures in Android MalwareFamilies," Expert Systems with Applications: An International Journal, vol. 41, no. 4, pp. 1104-1117, 2014.

[4] H. L. Thanh, "Analysis of Malware Families on Android Mobiles: Detection Characteristics Recognizable by Ordinary Phone Users and How to Fix It," Journal of Information Security, pp. 213-224, 2013.

[5] K. Aktas and S. Sen, "UpDroid: Updated Android Malware and Its Familial Classification".

[6] B. B. R. A. R. K. V. T. Raman Dugyala, "Application of Information Flow Tracking for SignatureGeneration and Detection of Malware Families," International Journal of Applied Engineering Research (IJAER), vol. 9, pp. 29371-29390, 2014.

[7] S. C. K. S. S. R. K. Lokesh Vaishanav1, "Behavioural Analysis of AndroidMalwareandDetection," International Journal of Computer Trends and Technology(IJCTT), vol. 47, pp. 176-181, 2017.

[8] F. M. V. N. S. a. C. A. V. Pasquale Battista, "Identification of Android Malware Families with Model Checking," in SCITEPRESS – Science and Technology Publications, Benevento, Italy, 2016.

[9] A. B. A.J.Singh, "Android Security: An Overviewof Risks and Security Measures," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 5, pp. 192-198, 2015.

[10] Y. Z. X. Jiang, "Dissecting Android Malware: Characterization and Evolution," in SP '12 Proceedings of the 2012 IEEE Symposium on Security and Privacy, Washington DC, 2012.

[11] M. A. S. K. M. A. Saba Arshad, "Android Malware Detection & Protection: A Survey," International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, pp. 463-475, 2016.

[12] G. M. D. B. B. M. Ashwini Mujumdar, "Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches," International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) , vol. 2, no. 6, pp. 2037-2039, 2013.

[13] A. M. Sweeney, "Malware Analysis & Antivirus Signature Creation," Quality and Qualifications Ireland (QQI), 2015.

[14] M. Chowdhury and K. Nygard, "Machine Learning within a Con Resistant Trust Model", The 33rd International Conference on Computers and their Applications (CATA 2018), March 19-21, 2018.

[15] M. Chowdhury, K. Nygard, K. Kambhampaty and M. Alruwaythi, "Deception in Cyberspace: Performance Focused Con Resistant Trust Algorithm", The 4th Annual Conference on Computational Science & Computational Intelligence, December 2017.

[16] M. Chowdhury and K. Nygard, "An Empirical Study on Con Resistant Trust Algorithm for Cyberspace", the 2017 World Congress in Computer Science Computer Engineering & Applied Computing, July 17-20, 2017.

[17] M. Chowdhury and K. Nygard, "Deception in Cyberspace: An Empirical Study on a Con Man Attack", The 16th Annual IEEE International Conference on Electro Information Technology, May 14-17, 2017.

[18] M. Chowdhury, “Deception in Cyberspace: Con-Man Attack in Cloud Services,” Ph.D. dissertation, Dept. of Computer Science, North Dakota State Univ., Fargo, ND, 2018. Accessed on: March. 25, 2020. [online]. Available: https://library.ndsu.edu/ir/handle/10365/28761

[19] R. Gomes, M. Ahsan and A. Denton, "Random Forest Classifier in SDN Framework for User-Based Indoor Localization", the 2018 IEEE International Conference on Electro/Information Technology.

[20] M. Ahsan, R. Gomes and A. Denton, "SMOTE Implementation on Phishing Data to Enhance Cybersecurity", the 2018 IEEE International Conference on Electro/Information Technology.

[21] R. Gomes, A. Denton and D. Franzen, "Comparing classification accuracy of NDVI with DEM derived attributes using multi-scalar approach in Geographic Information Systems," 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 249-254,

[22] M. Ahsan, R. Gomes and A. Denton, "Application of a Convolutional Neural Network using transfer learning for tuberculosis detection," 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 427-433.

[23] M Ahsan, K Nygard, "Convolutional Neural Networks with LSTM for Intrusion Detection," Proceedings of 35th International Conference on Computers and Their Applications, vol 69, pages 69--79.

[24] M. Chowdhury, J. Tang and K. Nygard, "An Artificial Immune System Heuristic in a Smart Grid", the 28th International Conference on Computers and Their Applications, 2013.

[25] M. Chowdhury, “An Artificial Immune System Heuristic in a Smart Electrical Grid,” M.S. Thesis, Dept. of Computer Science, North Dakota State Univ., Fargo, ND, 2014. Accessed on: March. 25, 2020. [online]. Available: https://library.ndsu.edu/ir/handle/10365/27236.