5-2019

Attacks on the Android Platform

Ranajit Singh Mehroke
St. Cloud State University, ranajit81@hotmail.com

Attacks on the Android Platform

by

Ranajit Singh Mehroke

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

in Information Assurance

March, 2019

Starred Paper Committee:
Susantha Herath, Chairperson
Jim Chen
Changsoo Sohn

Abstract

The focus of this research revolves around Android platform security, specifically Android malware attacks and defensive techniques. Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. With the rise of device mobility in our data-driven world, Android constitutes most of the operating systems on these mobile devices playing a dominant role in today's world. Hence, this paper analyzes attacks and the various defensive mechanisms that have been proposed to prevent those attacks.

Keywords: Android, Malware, Security, Defensive Mechanism, Mobile Devices

Acknowledgements

I want to thank the Information Assurance department for their continuous support of my Starred paper and research. Besides the department, I would like to thank my advisor Dr. Susantha Herath, and my committee members, Dr. Jim Chen and Dr. Changsoo Sohn.

Table of Contents

List of Figures

Figure Page

Figure Page

Chapter I: Introduction

Smartphones and tablets have become a part of our daily lives so much so that we are dependent on it to be connected to the rest of the world through the World Wide Web. There are various brands and models utilizing a variety of mobile operating systems (OS); each one tailored to suit the user needs. At the end of 2018, Android has been the best-selling OS worldwide on smartphones and tablets. The rest of the market share is made up by Apple's IOS, Microsoft's Windows Mobile, Blackberry's RIM OS, and Symbian OS. Symbian was the most popular mobile OS until Android overtook it in 2010.

Android has a Linux foundation which has been used primarily for touchscreen mobile devices such as smartphones and tablets. The main difference between Android and other mobile OS's is the user experience, speed, web browsing, mobile payments and security implemented in the devices.

Popularity and the majority usage makes it a target for constant malicious attacks. There would be almost no resistance to hackers gaining access to all devices until a security patch is issued by the device manufacturer which could take months depending on their ability to track and plug the loophole.

Android is the current 'king' in the mobile device market. Unlike other mobile OS which is proprietary, Android is open source making it easier to understand and perform malicious attacks on its coding structure.

This paper should familiarize the reader to the types of malware attacks which can invade Android's secure platform and suggests some prevention techniques and best practices to overcome it.

Problem Statement

The widespread use of the Android operating system has made it a target of malicious applications which would lead to vulnerabilities in Android platform and applications. Malicious apps are targeting the Android operating system to gain access and privileges to personal information. This information includes text messages, photos, contact, banking, and location information. Mobile devices with video capability enable monitoring of users to further infringe on their privacy. Users' technical understanding may be limited to counter these attacks but understanding the attacks would prove beneficial in its prevention.

Nature and Significance of the Problem

This research could provide information on the issues on the recent Android versions particularly on the integrity, vulnerability, and security of the operating system. It would be a review on the current security threats facing the mobile platform which has gained significant market share over the years. It would be beneficial to the communities of Android developers and tech-savvy users as it would alert them to the potential bugs in the platform. Furthermore, this research would provide the necessary information on the different threats and attacks related to the flaws in the system. To future researchers, this research can provide baseline information on the risks they are facing.

Objectives of the Research

To conduct a comprehensive review of the vulnerabilities and attacks on the Android platform as well as to identify the security loopholes. It will also implore users to protect their Android devices. This research is an investigation of all the attacks unique on the Android platform and will recommend a set of remedies and preventive techniques.

Expected Outcomes

To raise awareness among developers and users alike that security comes hand in hand with new hardware and software. Although it is mostly up to developers and manufacturers to implement the necessary security measures, users play the most crucial role in recognizing social engineering methods which is much favored among attackers as it can easily circumvent all implemented technical measures.

Summary

The chapter introduces the reasoning behind the research into raising awareness on the vulnerabilities of the Android platform. Non-technical methods like social engineering can easily bypass established security measures which show there is a need for user training to understand the Android platform to protect them from such attacks.

Chapter II: Background and Review of Literature

This chapter covers a general evaluation of malware attacks and their impacts on the Android operating system, as well as different types of attacks that are possible. Various publications, journals, online news articles, and relevant books have been consulted to meet those requirements.

Background Related to the Problem

Malware is short for malicious software, meaning software that can be used to compromise computer functions, steal data, bypass access controls, or otherwise cause harm to the host computer. Malware is a broad term that refers to a variety of malicious programs. The most common types of malware; adware, bots, bugs, ransomware, rootkits, spyware, viruses, worms and phishing (Adware, n.d.). Below are the common types of malware in more detail description.

Adware. Adware is a software that generates revenue for its developer by producing online advertisements in the user interface of the software or on a screen presented to the user during the installation process. The software will generate two types of revenue:
One is for the display of the advertisement and the other on a "pay-per-click" basis if the user clicks on the advertisement pop-up. The software's functions could also be designed to analyze the user's location, and internet sites visited and to present advertising pertinent to the types of goods or services featured there (Adware, n.d.).

Advertising functions are integrated or bundled with the program in legitimate software. The developer uses adware to recover development costs and to generate revenue. In some cases, the developer may provide the software to the user free of charge or at a reduced price. The

income derived from presenting advertisements to the user may allow or motivate the developer to continue to develop, maintain and upgrade the software product (Adware, n.d.).

The term adware is mostly used to describe a form of malware which presents unwanted advertisements to a user. The ads produced by this malicious ‘adware’ are sometimes in the form of a pop-up or sometimes in a window that cannot close. Typically, it uses a deceitful method to either disguise itself as legitimate or piggybacks on another program to trick the user into installing it on their device (Adware, n.d.).

While some sources rate adware only as an "irritant," others classify it as an "online threat" or even rate it as seriously as a virus. Malicious adware does not give any notifications that it is gathering information (What is Adware?, n.d.). Adware has also been discovered in certain low-cost Android devices, particularly those made by small Chinese firms (Adware, n.d.). Here are a few typical signs that a device might contain adware:

- Advertisements appear in places they shouldn’t be.
- The web browser’s homepage has mysteriously changed without user permission.
- The web pages typically visited are not displaying correctly.
- The web browser performance slows to a crawl.
- The web browser crashes.

Source: Adware, n.d.

There are two main ways in which Adware can get onto an Android device:

- Freeware or shareware - Adware can be included within some freeware or shareware programs as a legitimate way of generating advertising revenues that help to fund the development and distribution of the freeware or shareware program.
- Infected websites - A visit to a contaminated site can result in the unauthorized installation of Adware on the device.

Source: What is Adware?, n.d.

Bots. Bots or Internet robots is a software application that runs automated tasks (scripts) over the Internet. Typically, bots perform functions that are both simple and structurally repetitive, at a much higher rate than would be possible for a human alone. While they may be utilized to perform repetitive jobs, such as indexing a search engine, they often come in the form of malware (Internet bot, n.d.).

The botnet is a collection of Internet-connected computers, each of which is running one or more bots. A 'Botmaster' controls the group of compromised computers. Botnet operators can use the aggregated power of many bots to raise the impact of those dangerous activities exponentially. A single bot might not be a danger for the Internet, but a network of bots indeed can create huge malfunctioning (Vania, Meniya, & Jethva, 2013).

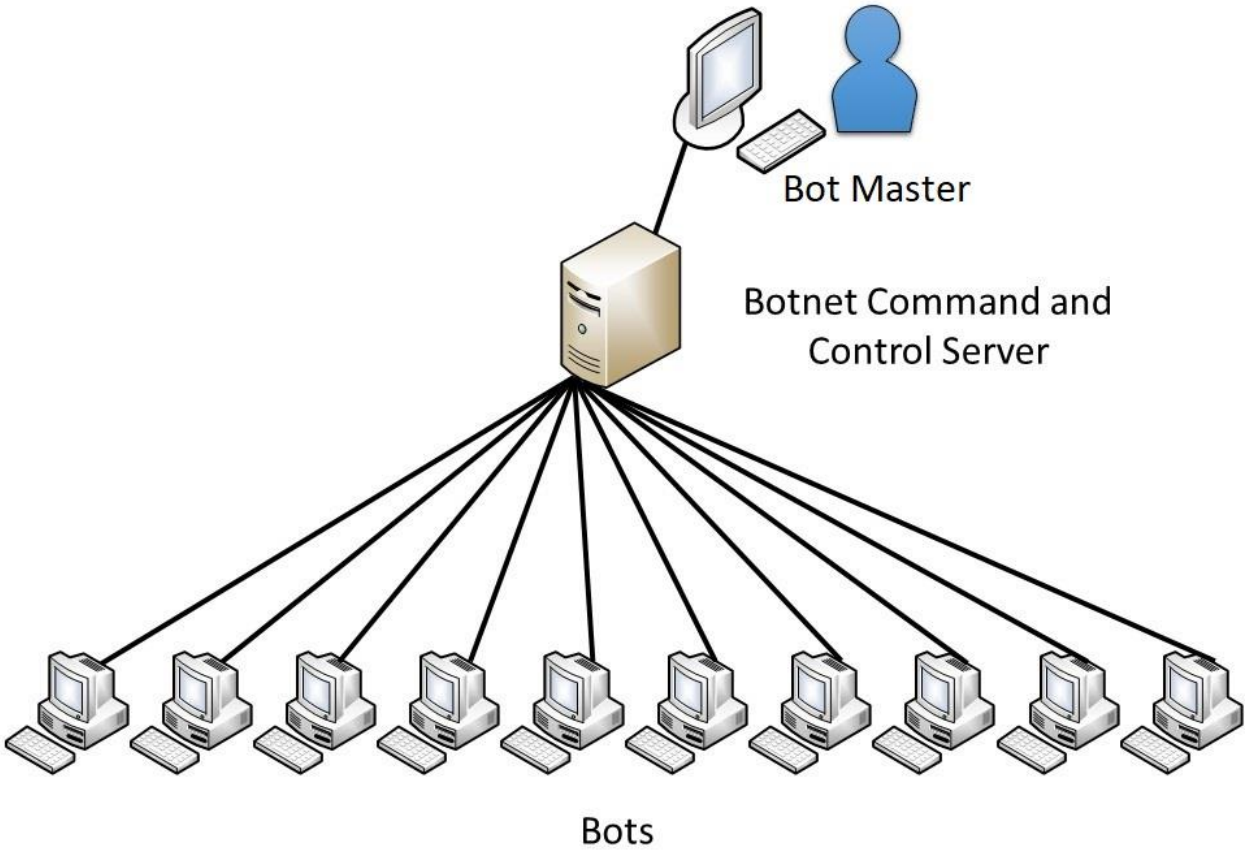

Figure 1: Botnet structure

Malicious bots have the "worm-like ability to self-propagate," and can log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch Distributed Denial of Service (DDOS) attacks, relay spam, and open backdoors on the infected host. They have been known to exploit backdoors opened by worms and viruses, which allows them to access networks that have good perimeter control. Bots rarely announce their presence with high scan rates that damage network infrastructure; instead, they infect networks in a way that escapes immediate notice (What Is the Difference: Viruses, Worms, Trojans, and Bots?, 2018).

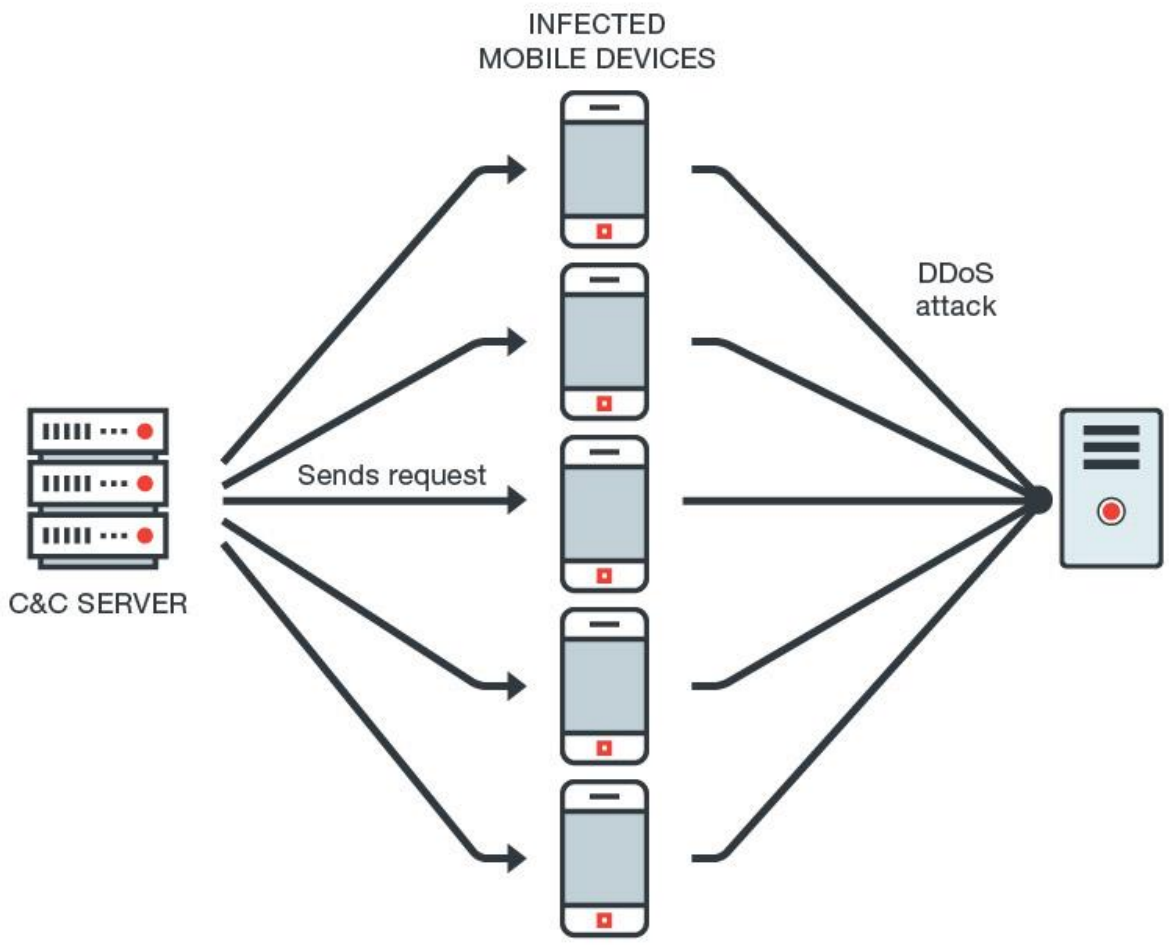

Figure 2: Botnet performing a DDOS attack

Below are the five most common malicious bots known so far:

- SPAM and SPIM bots - These bots bombard the user's inbox with SPAM and interrupt chats by sending unsolicited instant messages (SPIM). Some unscrupulous advertisers use these bots to target individuals based on demographic information obtained from the user's profile.
- Zombie Bots - A zombie bot is a device that has been compromised and has become a slave to the person who controls it along with hundreds or thousands of other devices as

part of a botnet. They use these zombie devices to coordinate large-scale attacks where all the zombie devices act in unison, carrying out commands sent by the master botnet owner. These infections can be difficult to detect and eradicate.

- File-sharing Bots - These bots take the user's query term (a movie or song title) and respond to the query stating that they have the file available and provide a link to it. In reality, the bot takes the search query term, generates a file by the same name (or similar name), and then injects a malicious payload. The unsuspecting user downloads it, opens it, and unknowingly infects their device.
- Chatterbots - Dating service websites and other similar sites are havens for malicious chatterbots. These chatterbots pretend to be a person and are generally good at emulating human interactions. Some people fall for these chatterbots, not realizing that they are malicious programs that attempt to obtain personal information and even credit card numbers from unsuspecting victims.
- Fraud Bots - These bots are more like scripts that try to achieve financial gain for their creators by generating false clicks for advertisement revenue programs, creating fake users for sweepstakes entries and generating thousands of fraudulent votes.

Source: O'Donnell, 2018

One of the many unfortunate things about malicious bots is the fact that they can quickly go unnoticed. They hide in “the shadows” of a computer, and many times have file names and processes similar if not identical to regular system files/processes. Below are some ways to tell if bots infect a device:

- Internet access is slow for no apparent reason.
- Device crashes for no apparent reason.

- Heats up when the device is idle.
- Takes a long time to shut down or fails to shut down correctly.
- Pop-up windows and advertisements appear even when the browser is not in use
- Friends and family receive email messages that were not sent by the user
- Device programs are running slowly.
- Device settings have changed, and there's no way to reverse them.
- The browser features components user did not download.

Source: What are bots?, n.d.

Bugs. A bug is a flaw which produces an undesired outcome. They are typically the result of human error and can exist in the source code of a program. There are different severity levels of bugs. Minor bugs only affect a program's behavior slightly, which is why they can go for long periods without being discovered. More significant ones can cause issues like freezing or crashing. Security bugs are the most severe of all; they can allow attacks to bypass user authentication, override access privileges or steal data (What is malware? A guide to the 12 most common types, 2018).

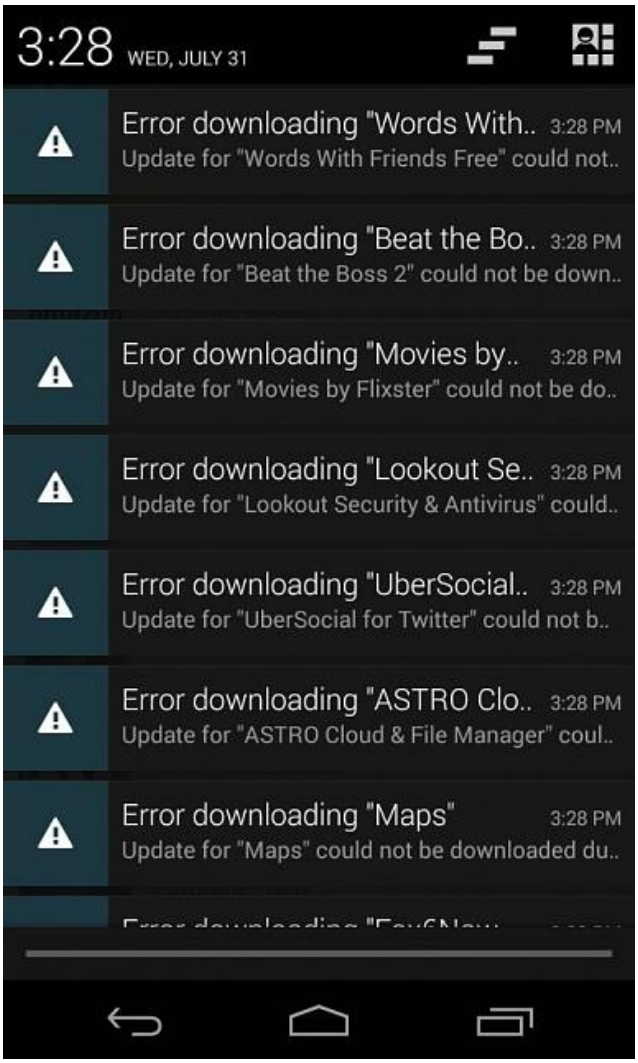

Figure 3: Bugs in Android 4.3 on a Nexus 4 phone

Recent versions of Android enable the user to send a bug report after the device experiences a crash related to the operating system or applications. A bug report contains device logs, stack traces, and other diagnostic information to help developers find and fix bugs in the app. This action is automatic from personal experience, but there is the **Take bug report** developer option on the device which enables the user to do so voluntarily. Android version 4.2 and higher support a Developer Option for taking bug reports and sharing it through e-mail (Capture and read bug reports, n.d.).

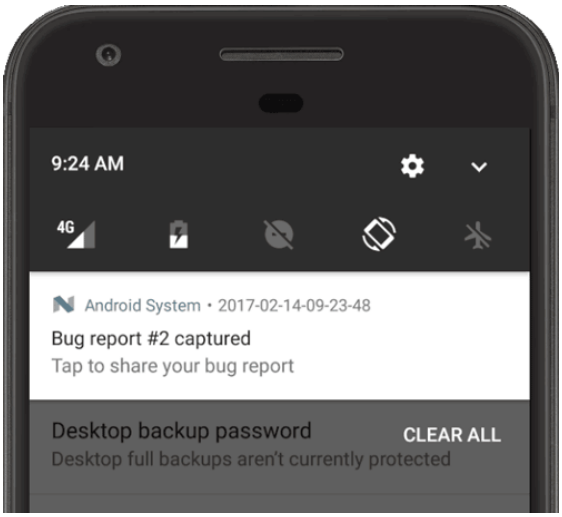

Figure 4: Android bug report

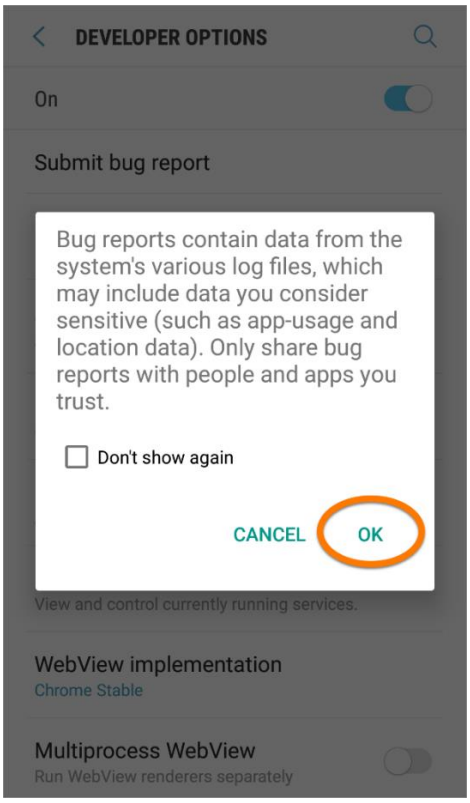

Figure 5: Option to enable Bug Report in developer options

After the submission of the bug report, the process is as follows:

1. A bug is filed and has the state "New."

2. An Android Open Source Project (AOSP) maintainer periodically reviews and triages the bugs into one of four buckets: New, Open, No-Action, or Resolved.
3. Each bucket includes several states that provide more detail on the fate of the issue.
4. A future release of the Android software will include bugs marked as "Resolved."

Source: Life of a Bug, n.d.

Android contains a lot of software and gets a correspondingly large number of bugs. As a result, sometimes bugs don't make it through all the states in a formal progression. Periodic "bug sweeps" are conducted where the database is reviewed and updated (Life of a Bug, n.d.).

Ransomware. Ransom malware, or ransomware, is a type of malware that prevents users from accessing their system or personal files and demands ransom payment to regain access (Ransomware - What is it & how to remove it, n.d.).

Over the years, the options and preferred methods of payment have changed as different services became available. In 1989, the AIDS crypto ransomware Trojan demanded payment by way of a check sent to a post office box in Panama. Other payment methods include wire transfers and sending premium-rate text messages as well as the use of payment voucher systems such as Paysafecard, MoneyPak, UKash, CashU, and MoneXy. The arrival of cryptocurrencies in the form of Bitcoin in 2009 made it easier for victims to purchase them to make ransom payments and then for the cybercriminals to convert them back into hard cash later (Savage, Coogan, & Lau, 2015).

Ransomware attacks are typically carried out using a trojan. A trojan horse or trojan is a type of malware that disguises itself as legitimate software (Kapersky, 2019). The user is tricked

into downloading or opening the trojan when it arrives as an e-mail attachment (Ransomware, n.d.).

One of the universal techniques used by trojans is obtaining Device Administrator privileges. Device Administrator privileges are not the same as root access, which would be even more dangerous if acquired by trojans. Legitimate Device Administrator applications use these extended permissions for various (mostly security-related) reasons. Trojans use this Android feature for its protection against uninstallation. Before such an app can be uninstalled, its Device Administrator rights must first be revoked (Lipovsky & Branisa, 2015).

Below is an example of ransomware which stays silent for the first four hours after it is installed, allowing the original app to operate without any interference. Once the app is running, this technique enables the ransomware to evade antivirus engines. After four hours, users will see a prompt to add a device administrator as shown in the figure below. If the user presses the 'Cancel' button, the prompt reappears, preventing the user from taking any other action or uninstalling the app. As soon as the user presses the 'Activate' button, the screen will be locked, and a full-screen ransom note will be displayed (Shinde, 2017).

Figure 6: Device Administrator prompt

There are two primary forms of ransomware in circulation:

- Locker ransomware (computer locker): Locker ransomware is designed to deny access to computing resources. It typically takes the way of locking the computer's or device's user interface and then asking the user to pay a fee to restore access to it. Locked computers will have limited capabilities, such as only allowing the user to interact with the ransomware and pay the ransom. It means access to the mouse might be disabled, and the keyboard functionality might be limited to numeric keys, allowing the victim to only type numbers to indicate the payment code.
- Crypto ransomware (data locker): This type of ransomware is designed to find and encrypt valuable data stored on the computer, making the data useless unless the user obtains the decryption key.

Source: Savage, Coogan, & Lau, 2015

Academic organizations, especially colleges and universities, have been among the top ransomware targets. Smaller IT teams, budgetary constraints, and a high rate of network file sharing are among the reasons educational organizations are vulnerable. Other ransomware targets include government agencies because the services they offer, such as police protection, are time-sensitive and crucial. Because such agencies often need to respond quickly, they have a greater sense of urgency in recovering their data and thus may be more willing to pay the ransom under duress. Hospitals may pay the payment because their patient data is critical in life-or-death situations (Fruhlinger, 2018).

Rootkit. A rootkit is a collection of computer software designed to enable access to a computer or areas of its software that is not otherwise allowed and often masks its existence or the existence of other software. Rootkit installation can be automated, or an attacker can install it after having obtained root or Administrator access. Once installed, it becomes possible to hide the intrusion as well as to maintain privileged access by gaining root or administrator access. Full control over a system means that existing software can be modified, including software that might otherwise be used to detect or circumvent it (Rootkit, n.d.). Rootkits once activated could be used to track the location of the mobile phone's owner, read their private SMS messages, and redirect calls to bogus numbers (Cluley, 2012).

Figure 7: Example of rootkit masquerading as Antivirus software

Older antivirus programs often struggled to detect rootkits, but most antimalware programs today could scan for and remove rootkits hiding within a system. One common symptom of a rootkit infection is that antimalware protection stops working. An antimalware application that stops running indicates that there is an active rootkit infection. Another sign of a rootkit infection is when device settings change independently, without any apparent action by the user. Unusually slow performance or high CPU usage and browser redirects may also indicate the presence of a rootkit infection (Rouse, 2018). Below are the different types of rootkit:

- Kernel rootkit - This type of rootkit is designed to function at the level of the operating system itself. What this means is that the rootkit can effectively add new code to the OS, or even delete and replace OS code. Kernel rootkits are advanced and sophisticated pieces of malware and require advanced technical knowledge to create one properly.

- Hardware or firmware rootkit - Instead of targeting the OS, firmware/hardware rootkits go after the software that runs specific hardware components. In 2008, a European crime ring managed to infect card-readers with a firmware rootkit. It then allowed them to intercept the credit card data and send it overseas.
- Hypervizor or virtualized rootkit - Virtualized rootkits are a new development that takes advantage of new technologies. A kernel rootkit will boot up at the same time as the operating system, but a virtualized rootkit will boot-up first, create a virtual machine and only then will it boot up the operating system. Virtualized rootkits have even more control over the system than a kernel one. And because they bury themselves so deep within the device, removal can be nearly impossible.
- Memory rootkit - Memory rootkits hide in the RAM of the device. Like kernel rootkits, these can reduce the performance of RAM, by occupying the resources with all the malicious processes involved.
- User-mode or application rootkit - User-mode rootkits are more straightforward and more natural to detect than kernel or boot record rootkits. It is because they hide in the application, and not system critical files. They operate at the level of standard programs which means a good antivirus or anti-rootkit program will probably find the malware and then remove it.

Source: Cucu, 2019

In 2015, there was a new malware campaign that used rootkits named Gooligan which had breached the security of over one million Google accounts. The number continues to rise at an additional 13,000 breached devices each day. By late 2015, it had gone mostly silent until the

summer of 2016 when it reappeared with a more complex architecture that injects malicious code into Android system processes (More Than 1 Million Google Accounts Breached by Gooligan, 2017).

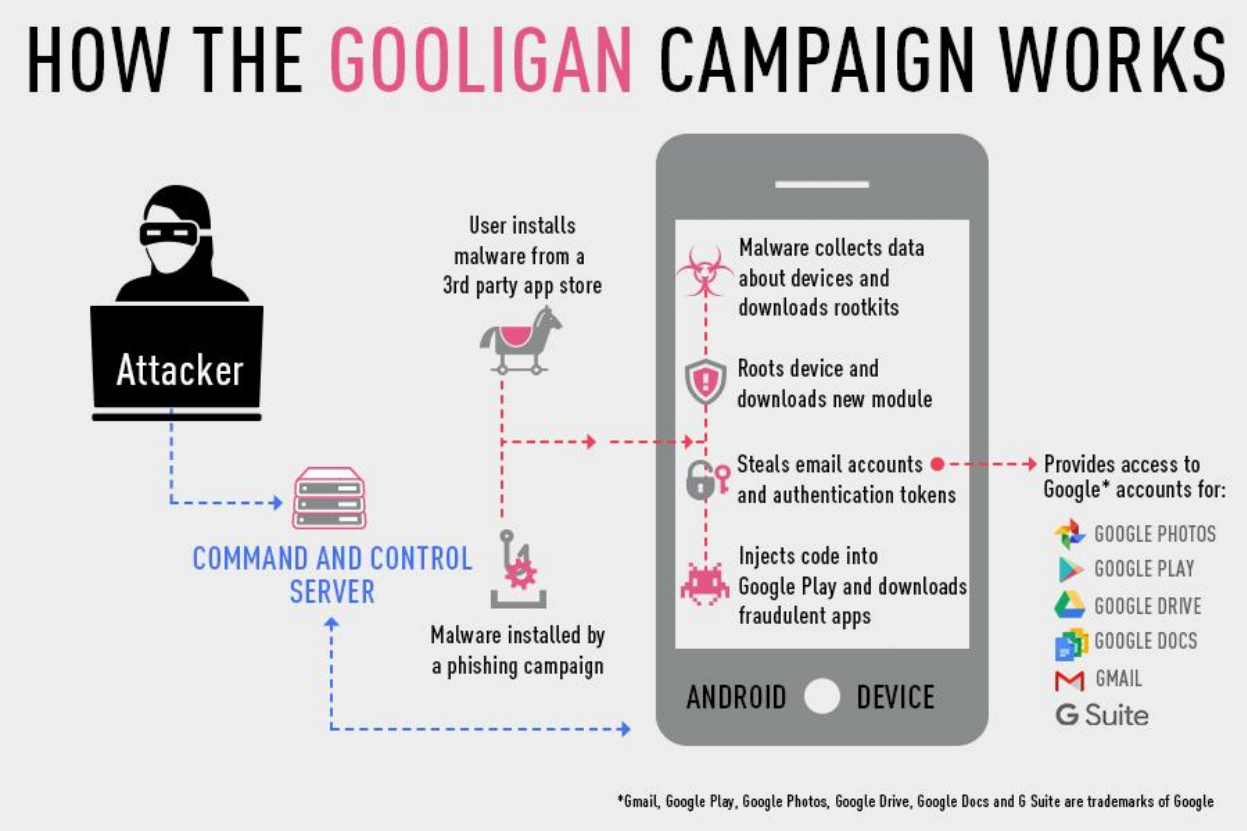

Figure 8: Gooligan campaign

The infection begins when a user installs a Gooligan-infected app on a vulnerable Android device. Gooligan then downloads a rootkit from the Command and Control (C&C) server that takes advantage of Android 4 (Jelly Bean, KitKat) and 5 (Lollipop) exploits. After achieving root access, Gooligan downloads a new, malicious module from the C&C server and installs it on the infected device. These exploits still plague many devices today because security patches that fix them may not be available for some versions of Android, or the user never

installed it. If rooting is successful, the attacker has full control of the device and can execute privileged commands remotely (Vegas, n.d.).

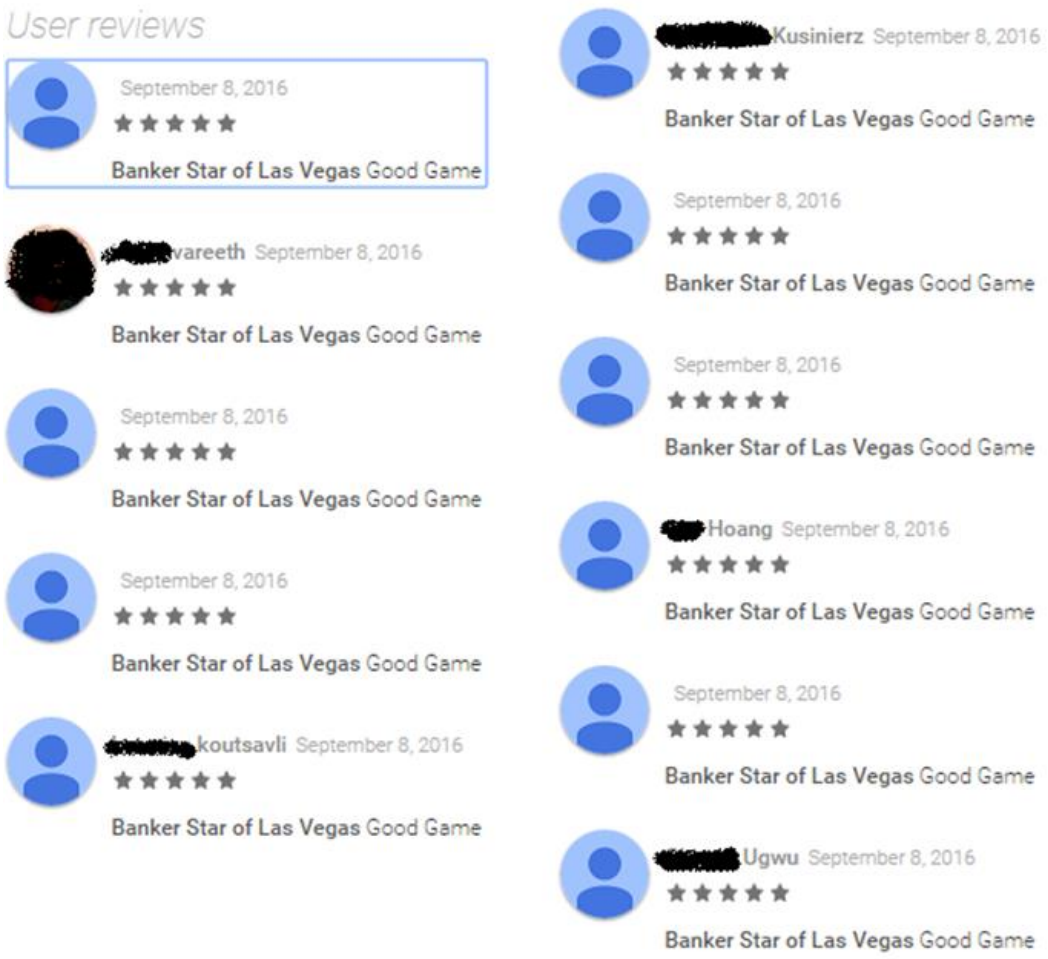

Figure 9: Fake reviews to entice users to download Gooligan-infected apps

Spyware. Spyware is software that aims to gather information about a person or organization, sometimes without their knowledge, that may send such information to another entity without the consumer's consent, that asserts control over a device, or it may send such information with the consumer's consent, through cookies.

Spyware can infect your system in the same ways that any other malware does, utilizing a Trojan, a virus, worm, exploit, and different types of malware (Spyware, n.d.).

Some of the most common ways your computer can become infected with spyware include these:

- Accepting a prompt or pop-up without reading it first
- Downloading software from an unreliable source
- Opening email attachments from unknown senders
- Pirating media such as movies, music, or games

Source: What is spyware? And how to remove it., n.d.

Skygofree is spyware that uses a novel technique to silently infiltrate Android phones and siphon off WhatsApp messages which are supposedly one of the most advanced forms of malware targeting Google's operating system ever seen. Its capabilities include the usage of multiple exploits for gaining root privileges, a complex payload structure, and never-before-seen surveillance features. As of October 2017, it could record audio via the microphone when an infected device was in a specified location and could force a target device to connect to Wi-Fi networks controlled by the attacker (Brewster, 2018).

Trojans. A Trojan horse, or Trojan, is a type of malicious code or software that looks legitimate but can take control of a device. A Trojan's design is to damage, disrupt, steal, or in general, inflict some other harmful action on the user’s data or network (What is a Trojan? Is it a virus or is it malware?, n.d.).

Trojans misleads users of its real intent and generally spread by some form of social engineering. Unlike computer viruses and worms, Trojans usually do not attempt to inject themselves into other files or otherwise propagate themselves (Trojan horse (computing), n.d.).

Viruses can execute and replicate themselves. A Trojan cannot. A user must run Trojans. One form of Trojan malware has targeted Android devices specifically. It is called Switcher Trojan, and it infects users by attacking the routers on their wireless networks. Cybercriminals could redirect traffic on the Wi-Fi-connected devices and use it to commit various crimes (What is a Trojan? Is it a virus or is it malware?, n.d.). Once a Trojan has completed transfer, it can:

- Give the attacker backdoor control over the Android device.
- Record keyboard strokes.
- Download and install a virus or worm to exploit a vulnerability in another program.
- Install ransomware to encrypt the user's data and extort money for the decryption key.
- Activate the device's camera and recording capabilities.
- Turn the computer into a zombie bot that can be used to carry out click fraud schemes or illegal actions.
- Legally capture information relevant to a criminal investigation for law enforcement.

Source: Rouse, 2018

FakeInst is a trojan masquerading as an Android application whose purported use is to watch pornographic films. After a user downloads, installs and opens the app, it then prompts that person to send a text message to purchase paid content from the app. In addition to its status as the premiere SMS Trojan targeting Android users in the U.S., FakeInst also focused on Android users in an additional 65 countries. As Kaspersky Lab notes, this is a mobile threat with

global ambitions. FakeInst is known to have targeted users in Germany, France, Finland, Hong Kong, Ukraine, the U.K., Switzerland, Argentina, Spain, Poland, Canada, China, and many more nations. Cybercriminals in Russia most likely developed the malicious application which can also intercept, delete, and even respond to incoming text messages (Donohue, 2014).

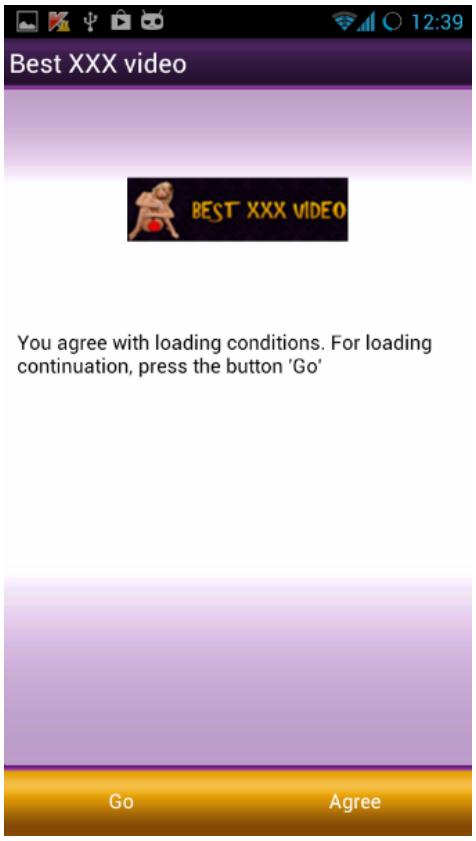

Figure 10: FakeInst app

Virus. A computer virus is a malicious program that self-replicates by copying itself to another program. In other words, the computer virus spreads by itself into other executable code or documents. The purpose of creating a computer virus is to infect vulnerable systems, gain admin control and steal user sensitive data. Hackers design computer viruses with malicious intent and prey on online users by tricking them (Judge, 2018). Once a virus has successfully attached to a

program, file, or document, the virus will lie dormant until circumstances cause the computer or device to execute its code. For a virus to infect a device, the user has to run the infected program. (What is a computer virus?, n.d.). Below are some common ways to get infected with a computer virus:

- Email viruses - Email is one of the favorite means of transportation for computer viruses everywhere. It is possible to get computer viruses through email by:
    - Opening an attachment - Often named as something harmless (such as “Your flight itinerary”), an executable program file (.com, .exe, .zip, .dll, .pif, .vbs, .js, .scr) or macro file type (.doc, .dot, .xls, .xlt, xlsm, .xsltm…).
    - Opening an email with an infected body - In these days of vibrant graphics and colors and bells and whistles, the HTML body of the email itself transports some viruses. Many email services disable HTML by default until you confirm you trust the sender.
- Instant messaging viruses - Instant messaging (IM) is another means for viruses to spread. Skype, Facebook Messenger, Windows Live Messenger, and other IM services are inadvertently used to send infected links to user contacts through chat messages. These instant messaging and social media viruses spread wide and fast because it is far easier to get people to click on a link coming from someone they trust, as opposed to an email from a stranger.
- File sharing viruses - Peer-to-peer file sharing services like Dropbox, SharePoint or ShareFile can be used to propagate viruses too. These services sync files and folders to any computer linked to a specific account, so when someone (inadvertently or otherwise)

uploads a virus-infected file to a file-sharing account, that virus gets downloaded to everyone else with access to that shared folder. Some file-sharing services, such as Google Drive, scan uploaded files for viruses (although it only scans files smaller than 25MB, giving virus spreaders an easy out — they have to make sure their virus-infected files are more substantial than that). But most other services do not scan for viruses at all, so it is the responsibility of the user to protect themselves against any potential threats contained in the downloaded file.

- Software download viruses - Fake antivirus infections are one of the most common types of virus-loaded software downloads. Scammers and cybercriminals use aggressive pop-ups and ads to scare users into believing that a non-existent virus has been detected in their device and compels them to download their "antivirus" software to clear the threat. Instead of ridding the computer of viruses, this fake antivirus proceeds to infect the device with malware, often with devastating consequences for the victim's files, hard drive, and personal information.
- Unpatched, vulnerable software - One of the most common means for viruses to spread is unpatched software. Unpatched software refers to software and apps which does not have the latest security updates from the developer, to plug up security holes in the software itself. Unpatched software is a significant cybersecurity headache for businesses and organizations, but with criminals exploiting vulnerabilities in outdated versions of such popular programs as Adobe Reader, Java, Microsoft Windows or Microsoft Office, users are very much at risk of infection too.

Source: Torres, 2017

Below is a list with five of the most dangerous viruses as of 2016:

- Shedun - it is a well-known type of Android malware software that can automatically root the device, leaving it open to a stream of advertisements. It's repackaged with legitimate apps, and it has been found pre-installed on several Chinese instruments in the past. As it is difficult to remove it entirely, which includes factory-resetting the device the user will have to root and reflash the device depending on the model and firmware which is difficult for many users.
- Godless - It can be found on apps in the Play Store and will root the device when the screen switches off, so there is no way of knowing as it infects the device quietly and efficiently.
- Cloned & copied apps - Popular applications are always going to be an easy target for hackers and scammers, and clones are a great way to get files onto devices. A guide application for Pokemon Go has infected over 500,000 users. Users unwittingly gave them access to their information. The worst thing is the more downloads it has, the more trustworthy it becomes.
- Hummingbad - It can steal user information and download apps without user permission. Cybersecurity company Check Point claims that it has spread to over two million devices worldwide by July 2016. It infects the phone by initiating a download when the user visits a suspicious website on the browser.
- Gunpowder - Users will install this virus through third-party emulators for Nintendo consoles found outside the Google Play Store. There are several emulators available for a

price, so it is sometimes worth paying a little extra for that added peace of mind as it is a fraction of the cost of the games themselves.

Source: Milin-Ashmore, 2016

Figure 11: Emulator example

Worm. A computer worm is a standalone malware program that replicates itself to spread to other devices. Often, it uses a network to cover itself, relying on security failures on the target device to access it (Computer worm, n.d.). In addition to using a network to spread, below are additional ways worm uses to spread:

- E-mail - One of the most common ways for infections to spread is through e-mail spam. E-mail attachments remain favorite hiding spots. What may appear to be a useful work document or personal photo can be hiding malicious code, waiting to be released when

the user clicks a link or opens an attachment. The worm may replicate itself by emailing itself to everyone in the user's address book or automatically replying to e-mails in the inbox.

- Operating system vulnerabilities - Every operating system has it, and some worms are specially coded to take advantage of these weak points.
- Instant messaging - Worms can take on similarly deceptive forms in instant messaging software and take advantage of users who are probably not on high alert when using such services. In today's digital landscape, modern chat systems are just as vulnerable, with Facebook Messenger as a common infection point for worms such as Dorkbot, which spreads through an executable file disguised as a JPEG image.
- Smartphones - Every primary mobile operating are potentially vulnerable to worms as they all support HTML5-based mobile apps. One of the critical security flaws of HTML5 is that malicious code can easily be inserted into it, meaning that when a user launches an app, they could also be unwittingly executing a damaging program.

Source: Jareth, 2018

Worms are different from viruses. Worms can exist as standalone software, but a virus needs a host file before it can spread. Worms do not require host files or programs to propagate (Raymond, 2018).

In addition to wreaking havoc on a device's resources, worms can also steal data, install a backdoor, and allow a hacker to gain control over a device and its system settings. Below are the symptoms that a device has a worm:

- Hard drive space - When worms repeatedly replicate themselves, they start to use up the free space on devices.
- Speed and performance – If the device is little sluggish and some of the programs are crashing or not running correctly, then it could be a red flag that a worm is eating up the device's processing power.
- Missing or new files - One function of a worm is to delete and replace files on a device.

Source: What is a computer worm, and how does it work?, n.d.

Cyber-security analysts at Trend Micro have revealed a backdoor worm that stealthily controls several functionalities of infected Android devices. If this doesn't sound alarming enough, the researchers have further added that this vulnerability will continue to evolve while secretly recording audio or voice and send to the attacking server in an encrypted manner — the GhostCtrl backdoor masquerades as a legitimate app such as WhatsApp, and the famous Pokemon Go. It continues to install a malicious APK package under the hood when the user is running the app. The attackers will be able to retrieve all the data and take control of the device after execution by harnessing a range of commands without the user's acknowledgment. (Verma, 2017).

Phishing. Phishing is a fraudulent attempt to obtain sensitive information such as usernames, passwords, and credit card details by disguising as a trustworthy entity in electronic communication. Typically carried out by e-mail spoofing or instant messaging, it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site (Phishing, n.d.).

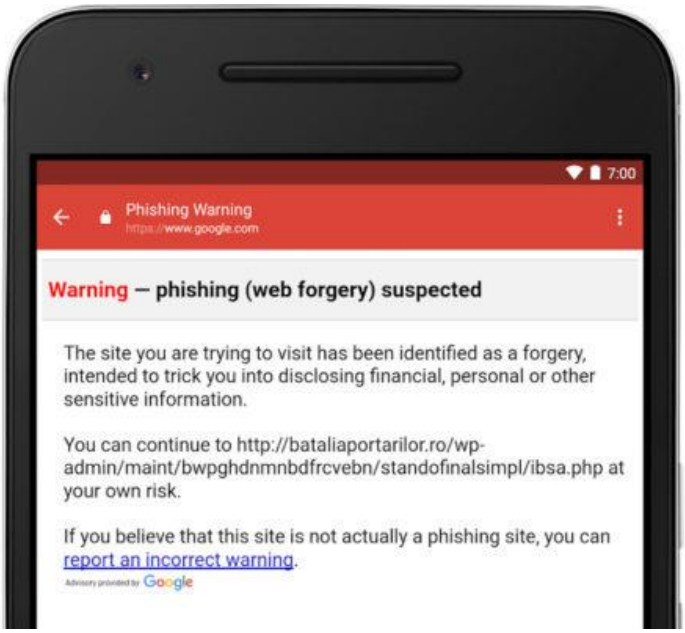

Figure 12: Gmail phishing application

Phishing is a criminal mechanism employing both social engineering and technical subterfuge. Below are the types of phishing schemes:

- Social engineering schemes use spoofed e-mails which are supposed to be from legitimate businesses and agencies, designed to lead consumers to counterfeit websites that trick recipients into divulging financial data such as usernames and passwords.
- Technical subterfuge schemes plant crimeware onto devices to steal credentials directly, often using systems to intercept consumers online account user names and passwords to misdirect consumers to counterfeit websites.

Source: Anti-Phishing Working Group (APWG), 2006a

Below are standard features of phishing e-mails:

- Too Good To Be True - Lucrative offers and eye-catching or attention-grabbing statements are designed to attract people's attention immediately. For instance, many

claims to have won an iPhone, a lottery, or some other lavish prize. If it seems too good to be true, then it probably is.

- A sense of Urgency - A favorite tactic amongst cybercriminals is to ask the user to act fast because the super deals are only for a limited time. Some of them will even say that the user has only a few minutes to respond. Sometimes, they will inform the user that they will suspend their account unless they update their details immediately. Most reputable organizations give ample time before they terminate an account and they never ask patrons to update personal information over the Internet.
- Hyperlinks - A link may not be all it appears to be. Hovering over a link shows that the actual URL where it will direct the user upon clicking on it. It could be completely different, or it could be a popular website with a misspelling, for instance, www.bankofarnerica.com - the 'm' is an 'r' and an 'n' so look carefully.
- Attachments - It is best that the user does not open any attachments from an unknown e-mail. They often contain payloads like ransomware or other viruses. The only file type that is always safe to click on is a .txt file.
- Unusual Sender - Whether it looks like it's from someone known or unknown, if anything seems out of the ordinary, unexpected, out of character or just suspicious then is it best not to click on it.

Source: What is phishing?, n.d.

It is not that difficult to create an Android phishing application. There are sites devoted to building these applications as shown below.

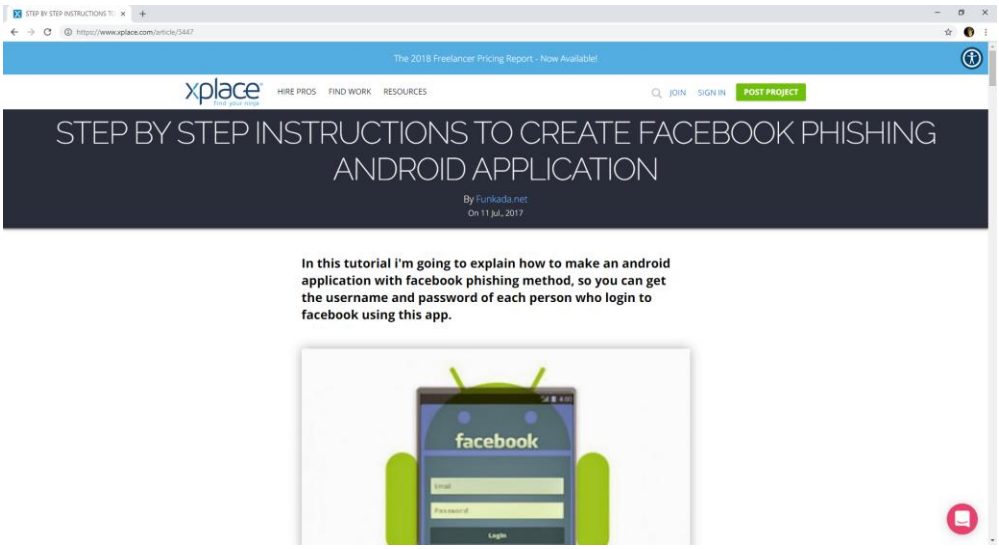

Figure 13: Instructions to create phishing app

Google had patched an Android flaw that fixes a security vulnerability in Android which allows phishing applications to spoof genuine ones. Security firm FireEye discovered that apps could modify the icons of other apps on Android home screens and make them point to any other app or website, which would allow attackers to divert users to fake versions of trusted apps and websites to steal information (Correspondent, 2014).

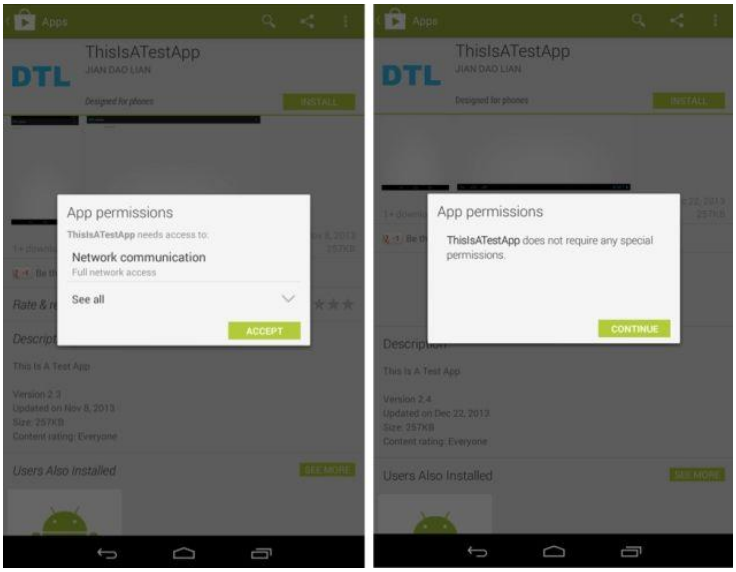

Figure 14: Android phishing flaw

Background and History

Android is a mobile operating system, initially developed by Android Inc. then sold to Google in 2005. It uses a modified Linux 2.6 kernel. Google, as well as other members of the Open Handset Alliance (OHA), collaborated on Android in its design, development, distribution. Currently, the Android Open Source Project (AOSP) is governing the Android maintenance and development cycle (Narmatha & KrishnaKumar, 2016).

Figure 15: Open Handset Alliance

In November 2007, 34 founding members established OHA dedicated to the development of open mobile standards, including Google, mobile device manufacturers, application developers, embedded systems developers, and commercialization companies. The goal of this alliance as described in the web site is as follows:

The Open Handset Alliance™, a group of 84 technology and mobile companies who have come together to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience.

OHA currently has 84 firms who are developing and working on the consortium's main and only project, Android. Thanks to the services and products offered by members of the OHA, higher quality for a lower price account for most devices and related services (Krajci & Cummings, 2013).

Android development involves many developers writing an application that helps in extending the functionality of the devices. There are over 1.6 million applications available for Android. The Play Store formerly known as Android Market is the online application store run by Google, though third-party sites do offer app downloads (Shah, 2016). Several drivers and libraries have been either modified or newly developed to allow Android to run as efficiently and as effectively as possible on mobile devices. Some of these libraries have their roots in open source projects. With Android, the focus has always been on optimizing the infrastructure based on the limited resources available on mobile devices (Krajci & Cummings, 2013).

Version History

There have been many updates of Android versions after the original release. These updates focus on fixing bugs as well as adding new features. Each new version has a code name based on a dessert item which is listed below:

- **Astro (1.0)** Astro was released as a beta in November 2007 and released to the public in September 2008 on the HTC Dream. Astro displayed various core features of the Android

OS and included many of the favorite apps such as Android Market, a web browser, e-mail/Gmail, Google Maps, Messaging, Media Player, YouTube, and various others.

- **Cupcake (1.5)** Released on April 30, 2009, Cupcake was based on the Linux kernel 2.6.27 and included many new features to users and developers. The significant changes were support for virtual keyboards, support for widgets on the home screen, animations added in various places, and auto-pairing and stereo support for Bluetooth-capable devices.
- **Donut (1.6)** Released on September 15, 2009, Donut came updated with a Linux kernel as well as some new features and supported devices. Major highlights included voice and text search of contacts/web/ bookmarks, support for WVGA screens, and improvements to camera functionality and speed. 'Donut' was the last version of Android in the 1.x series to be released.

  Some bugs and glitches detected in Donut 1.6. One issue was the Bluetooth connection to car kits. A potential solution involves rooting the device using a custom image to fix the issue. Another problem is the screen lock which stopped working. The user is unable to unlock their device to access the content and functions. It was recommended to uninstall installed applications one by one. This recommendation seemed to have worked for most of the users (Dibben, 2009).
- **Éclair (2.0/2.1)** Released October 26, 2009; it came with many new features and capabilities for both developers and consumers. Substantial changes to the way Android looked and felt included significant speed improvements in many different applications. The premier device for Android 2.0 was Motorola's Droid on Verizon Wireless. On

December 3, 2009, Google updated Android to version 2.0.1 in efforts to fix some small bugs and upgrade the API for developers. It wasn't until January 12, 2010, that Android updated to version 2.1. Similar to the update in December, version 2.1 primarily included upgrades to the underlying API and bug fixes.

- **Froyo (2.2.x)** On May 20, 2010, Google's Nexus One was the first device on the market to show off Froyo and its new capabilities. Very significant features included Adobe Flash support, Android Cloud to Device Messaging, Wi-Fi hotspot functionality, and significant performance optimizations. There were three subsequent updates for the Android 2.2 SDK: 2.2.1 on January 18, 2011, 2.2.2 on January 22, and 2.2.3 on November 21. These updates were primarily bug fixes and security updates to Android.

  Stagefright is the name given to a group of software bugs that affected version 2.2 ("Froyo"). Exploitation of the bug allows an attacker to perform arbitrary operations on the victim's device through remote code execution and privilege escalation. Security researchers demonstrated the bugs with a proof of concept that sends specially crafted MMS messages to the victim device, and in most cases, the user did not have to do anything to 'accept' the exploits because of the bug as it happens in the background (Stagefright (bug), n.d.).

- **Gingerbread (2.3.x)** Released on December 6, 2010, Google's Nexus S was introduced to show off Gingerbread. Features of Gingerbread include support for WXGA and other extra-large screen sizes, improvements to the virtual keyboard, support for more internal sensors (namely gyroscopes and barometers), support for front-facing cameras, and the ability to read Near Field Communication (NFC) tags. There were five updates for

Gingerbread; 2.3.3–7, from February to September of 2011. With these updates came various features, security updates, and bug fixes.

A computer security researcher at NC State University identified a security vulnerability in Gingerbread. By just clicking on a link, Android users gave attackers access to personal information. The vulnerability would allow a malicious web site to read and upload the contents of any file stored on the device's microSD (memory) card. Information on the SD card could include saved voicemails, photos or online banking data. The vulnerability would also allow attackers to find out all the applications installed on a device, and upload many of the applications onto a remote server. The vulnerability was confirmed using Gingerbread run on a Nexus S phone. A similar vulnerability was reported on earlier versions of Android phones, leading Google to make changes in Gingerbread design to address the flaw (Shipman, 2011).

- **Honeycomb (3.x)** Released in February of 2011, the first tablet-only Android version, was released on the Motorola Xoom. Because Honeycomb was created specifically for tablet devices, Android was tweaked to allow for a more enjoyable experience with larger screen real estate. It included a redesign of the onscreen keyboard, a system bar to allow for quick access to notifications and navigation, multiple browser tabs to allow for more comfortable use of the web, and support for multi-core processors. Honeycomb has had six updates, two of which were significant, through its current life cycle. The first update was Android version 3.1 on May 10, 2011, and it namely added support for USB accessories such as keyboards, joysticks, and other human interface devices (HIDs). The second major update was 3.2 on July 15, 2011. The most significant feature of 3.2 was

compatibility display mode for Android applications for tablets. The last four updates to Honeycomb have been minor improvements, bug fixes, and security updates. After extensive use across multiple tablets, researchers have identified weaknesses that Google needs to address in Honeycomb:

- Improve Image Rendering - Photos viewed in Android 3.0's Gallery application appeared fuzzy and washed out when compared to when seen on other devices. On the Honeycomb tablets-Motorola Xoom, T-Mobile G-Slate, Acer Iconia A500, and Asus Eee Pad Transformer TF101 the images lack sharpness and detail.
- Clean Up the Root Directory - Android lets the user have freer access to their files from within apps. It means the ability to transfer data to the tablet using email through an app or a memory card, or even a direct connection to the PC, and then access those files using another app on the device to do something with them. Many file manager apps will allow access to the Android file system, but by default, the file/folder organization looks cluttered. For example, on an Asus Transformer tablet, the file manager showed USB drive folders buried under the Removable directory, within the root directory.
- Improve Handling of External Storage - Many of the Honeycomb tablets have, at the least, a microSD card slot for expanding storage. And some have USB ports, either directly built into the tablet or made into a docking station that connects to the tablet. Unfortunately, Honeycomb appears ill-equipped for handling external storage. Users had to remove and reinsert media multiple times to get it to be

recognized. Android 3.0 gives a "force close" error, or not identifying the media plugged into that port or slot when the card is taken out without first unmounting.

- **Ice Cream Sandwich (4.0.x)** Released on October 19, 2011, Samsung's Galaxy Nexus was the device released with Ice Cream Sandwich (ICS) as it hit public markets. ICS included a multitude of features and improvements to the Android user interface (UI). Some highlights include a customizable launcher, a tabbed web browser, facial recognition to unlock the device, a built-in photo editor, hardware acceleration of the UI, and software buttons originally introduced in 3.x (Honeycomb). It is important to note that ICS merged version 3.x (Honeycomb) and 2.3.x (Gingerbread) into a single OS supporting both phones and tablets. Four minor updates have since been released for ICS devices from November of 2011 to March of 2012. These updates focused on stability improvements, camera performance, and bug fixes. The following is a list of bug reports which users have reported on the Android ICS (4.0.3):
    - GPS is not locking on the signal
    - GPS and battery functions are showing massive data usage
    - Contacts crashing and doubling in texts
    - Screen freezes during web browsing
    - Wi-Fi constantly scanning, disconnects and huge data downloads
    - Wi-Fi connection – Not connecting automatically to the known router while users are under the impression that they are downloading large data over Wi-Fi when it is using a mobile data connection.
    - Device freeze when unlocking to answer a call

- **Jelly Bean (4.1.x)** Jelly Bean was released on July 9, 2012, on the Asus Nexus 7 tablet device. Jelly Bean released several improvements and performance upgrades to the UI and audio within Android. Version 4.2, released on November 13, 2012, added accessibility improvements. Version 4.3 was released on July 24, 2013, and added OpenGL ES 3.0 support for better game graphics, security enhancements, and upgraded digital rights management APIs. Other features of the Jelly Bean versions include customizable keyboard layouts, expandable notifications, application-specific notification filtering, and multichannel audio.

  The following is a list of bug reports which users have reported on Jelly Bean:

  - Wi-Fi was dropping or not connecting - The common thread for users seems to be the update to Jelly Bean, but different version updates have impacted different devices. There are many potential reasons that Wi-Fi issues can occur, and it might be due to a specific router or the settings on it.
  - Random rebooting or freezing - Phone keeps freezing or crashing and rebooting itself after a Jelly Bean update which could be caused by an incompatible app. The issue is the software is updated, but the app hasn't. A bug could have also created it in Jelly Bean. Google released Jelly Bean 4.2.2, and it was supposed to fix several bugs including the random reboot issue.
  - Miserable battery life - Complaints about poor battery life are common for smartphones generally, but some users have complained that their device’s battery life has been noticeably worse after upgrading to Jelly Bean. It could be down to specific apps or services.

- Bluetooth is not working - Google has confirmed that there is a Bluetooth problem that can disrupt audio streaming. If the user has experienced issues with their Bluetooth functionality after upgrading to 4.2, then the bug is most likely the reason.
- Missing December - A high profile incident saw December missed out of the People app calendar on Nexus devices in the 4.2 updates.
- Lockscreen Widgets - Several of the lock screen widgets featured in Jelly Bean are suffering from glitches. The most noticeable being the Google Music widget. When a third-party media app is running the background, the widget's playback controls disappear. Some users have resorted to rebooting their device to have the program run correctly.

- **KitKat (4.4.x)** Released on September 3, 2013, its features included performance optimizations for devices with less RAM, expanded accessibility APIs, wireless printing capability, and a new experimental runtime virtual machine, called ART, which may come to replace Dalvik. KitKat debuted on the Google Nexus 5 smartphone on October 31. 2013 (Krajci & Cummings, 2013).

  The following is a list of bug reports which users have reported on KitKat:

  - Overheating - most likely caused by a third party application, such as Facebook, Viber, Skype and such.
  - Hardware and network unreliability - The device consumes high CPU on network hosts. This bug has been known to exert influence on users of Nexus 5. Android

4.4 holds many errors in network stability, and it carries low signal which causes problems for the users to make call or text.

  - Connectivity and battery affairs – Users, undergo timeout when discontinuation of connection occurs. Instability of internet connection turns out to update the WIFI software every time the user uses it.
  - Launcher issues - When users started Google search and typed on the search bar or when doing some editing on Gmail, the Google Play services will crash. It also occurs while downloading anything from Google Play as it will force close the applications.
  - Loss of packets - There's a high loss of packets when broadcasting data over a VPN connection. The apparent problem is that the Android 4.4 TCP protocol exhibit an incorrect "maximum segment size" for VPN packet transfer. The end consequences could be distorted piece of data and discontinuation of the network.

- **Lollipop (5.0-5.11)** One of the most prominent changes in the Lollipop release is a redesigned user interface built around Material Design as new a design language. Other changes include improvements to the notifications, which can be accessed from the lock-screen and displayed within applications as top-of-the-screen banners. The following is a list of bug reports which users have reported on KitKat:
  - Bluetooth is not working - There have been different reports about Bluetooth issues with Android 5.0 Lollipop. Some users are having trouble pairing devices, some can establish the connection, but not all the functions work correctly, and some report that it disconnects, apparently at random.

- Flashlight time out - Most users encountered a bug with the flashlight application which also impacts the camera. If the user turns the flashlight on through the quick settings toggle in the notification shade and allow it to time out after a few minutes, then it cannot be turned on again. The user may not be able to use their camera either.
- The camera is not working - Quite a few people have been having trouble with the camera after updating to Lollipop. Some people have also found that icons are disappearing, so the option to switch to the front-facing camera might not be there.
- Unable to close all applications - Some users might be in the habit of tapping the multitasking button and closing all their open apps, but that option to close them all is not available in Lollipop. Users can still swipe them away one-by-one.
- GPS is not working - Most users have found that the GPS performance in their phone has declined after updating to Lollipop. It may struggle to get a fix or get stuck for a few seconds before upgrading. Sometimes the accuracy might be off.
- Unable to connect to the mobile network - Some users have been having trouble connecting to their cellular network after updating to Lollipop. It may refuse to communicate at all, or it might connect intermittently.

Source: Shah, 2016

- **Marshmallow (6.0 – 6.0.1)** Android 6.0 gets better control over permissions, allowing the user to control what parts of data apps the user can access, rather than approve it by just installing the app in the first place. Features like app linking and the new Assist API

will allow developers to build better and more powerful apps (Narmatha & KrishnaKumar, 2016). The following is a list of bug reports which users have reported on Marshmallow:

- Wi-Fi battery drain - A lot of Marshmallow users have been posting screenshots of their battery usage page, which show Wi-Fi to be the most significant battery drain. In many cases, it is just false reporting, and the device's battery life is unaffected by Wi-Fi, but some other users are seeing the same data, and it is affecting their battery.
- Stability and performance issues - Not all updates are perfectly stable. Some of the bugs detected are random reboots, app crashes, glitches and stutters, strange behavior and so on.
- Charging problems - Despite Marshmallow's new USB charging standards, Android 6.0 charging problems seem familiar, from devices that drain faster to phones that won't charge at all. Marshmallow is making users earn the battery benefits of its Doze mode and app standby.
- Mobile data problems - The connection to mobile data behaves strangely after the update.
- Not connecting to PC - If the user is having USB connectivity issues, it is most likely because the right option is not selected phone's software. By default, the device will likely be set to USB charging mode, meaning the user won't be able to access your files.

- Insufficient storage available message - This issue is related to the SD card. Marshmallow does not allow SD cards to store apps. Most apps must remain on the actual internal storage.

Source: Marshall, 2016

- **Nougat (7.0 – 7.1.2)** Officially released on August 22, 2016, with the Nexus 6, 5X, 6P, 9, Nexus Player, Pixel C, Nougat displayed additional features which were supposed to be on Marshmallow. In addition to grouping notifications by app, which Marshmallow already did, there also drop-down menus. Split-screen comes standard on Android Nougat as well as the option of being able to close all open apps at one time. To close the apps on Marshmallow, you need to swipe each one to the side. Android Nougat implements Dark mode which allows the user to apply a filter to the screen to minimize eye strain and reduce the blue light emitted by the smartphone screen. Blue light harms sleep as it disturbs the production of melatonin, an essential sleep hormone (Vitre, 2018). The following is a list of bug reports which users have reported on Nougat:
  - Battery drain - It may seem counter-intuitive for an update which is meant to optimize battery usage to cause it to drain faster, but it can sometimes be the case.
  - Black screen issues - This bug was also present with Android Marshmallow and led to an unresponsive blank display.
  - Not connecting to PC - If the user is having USB connectivity issues after the Nougat update, it's most likely because the user did not choose the correct option in the settings. The default setting is USB charging mode, which means the user is unable to see files when plugged into the PC.

- Random rebooting problem - Several users have experienced boot loop and random rebooting issues with Nougat. The Nexus 5X seems to have a hardware issue with a small number of devices which causes boot looping.

Source: McGhee, 2017

- **Oreo (8.0 – 8.1)** Officially released on August 21, 2017, Sony Xperia XZ1 and Sony Xperia XZ1 Compact were the first devices available with Oreo pre-installed. Android 8.1 was released in December 2017 for Pixel and Nexus devices, which features minor bug fixes and user interface changes (Android Oreo, n.d.). Android Oreo allows activities to launch in picture-in-picture mode. Video playback uses this multi-window mode. Android Oreo also introduces the Autofill Framework. It is designed to make filling out forms, such as login and credit card forms, much faster and easier. It also supports Wi-Fi Aware, also known as Neighbor Awareness Networking or NAN. Devices with the right hardware can discover and connect directly to each other through Wi-Fi Aware. Clusters of neighboring devices make up for these networks. There is now enhanced support for multiple displays. If an app is running with multiple screens and activity supports multi-window mode then users can move the event from one display to another (Sims, 2017). The following is a list of bug reports which users have reported on Oreo:
    - Swipe to Unlock Issue - To unlock the phone the user must swipe for a long distance on your screen to open it. Some users claim that they must swipe to the edge of the screen to unlock the phone. While this bug was present in the Beta version too, it still frustrates some users.

  - Bluetooth/ Wi-Fi Connectivity Issue - On connecting to audio device wireless the phone will not automatically stream audio to Bluetooth speakers.
  - Display Issue, Ambient Display Not Working - The always on display feature on many devices with an AMOLED display is facing some problems after the 8.1 release. After receiving calls or notification, the ambient display on the handset get automatically turned off.
  - App Crash Problem and Errors - Some apps like the stock Mail and Clock app are seeing the random crash
  - Battery Drain After Oreo 8.1 Update - The latest release should improve battery performance, but some devices are seeing a drop in battery performance. Users reported the problem in the earlier Nougat version.

- **Pie (9.0)** Officially released on August 6, 2018; it was initially available for Google Pixel devices and the OnePlus 6. The Sony Xperia XZ3 was the first device with Android Pie pre-installed (Android Pie, n.d.). Below are some new features of Pie:
  - Dashboard - displays the time users spend on different applications as well as the overall time spent on the phone. It is meant to be a health feature which gets users into a more proactive approach towards using their smartphone devices. It is in the form of a pie chart showing the total time spent on the phone, broken down application wise (Srivastav, 2018).
  - App Timers - sets a time allowance for specific apps, one that resets at midnight each day. The user will get a reminder when that allowance is nearly gone. Once it has run out, the app icon will grey-out.

- Night Light - takes the blue light out of the display as it gets closer to bedtime. It is easier on the eyes, makes the screen look yellow/orange, and makes phone use less of a sleep disturbance.
- Wind Down - turns the display grey-scale and the 'Do Not Disturb' mode is activated as the Night Light feature reaches ‘bedtime.’ Like the other digital wellness feature above, Wind Down makes unhealthy phone use easier to avoid.
- Adaptive Brightness - is more intelligent. It learns from the tweaks made in different lighting conditions, mapping out a custom backlight curve. In theory, the Auto Brightness settings should land on exactly the level the user wants after a few days of regular use.

Source: Williams, 2018

Source: Shah, 2016

As Android Pie is relatively new, users should anticipate the same common bugs and glitches as previous versions after the update. More observation by users and developers alike will be needed to determine the authenticity of these bug reports.

Architecture

The Android OS is the Linux kernel in the form of a software stack comprising of applications, an operating system, run-time environment, middleware, services, and libraries. The modifications to the kernel for Android is a set of patches to the standard Linux kernel, which Google periodically upgrades to the latest released Linux kernel version (An Overview of the Android Architecture, 2017). Figure 16 shows the significant components of the Android architecture in further detail.

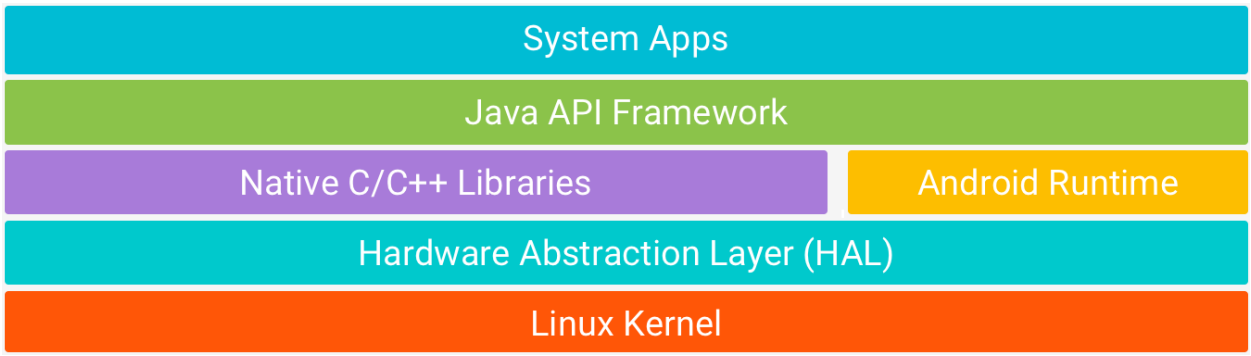

Figure 16: Android Architecture (Platform, 2018)

Linux Kernel

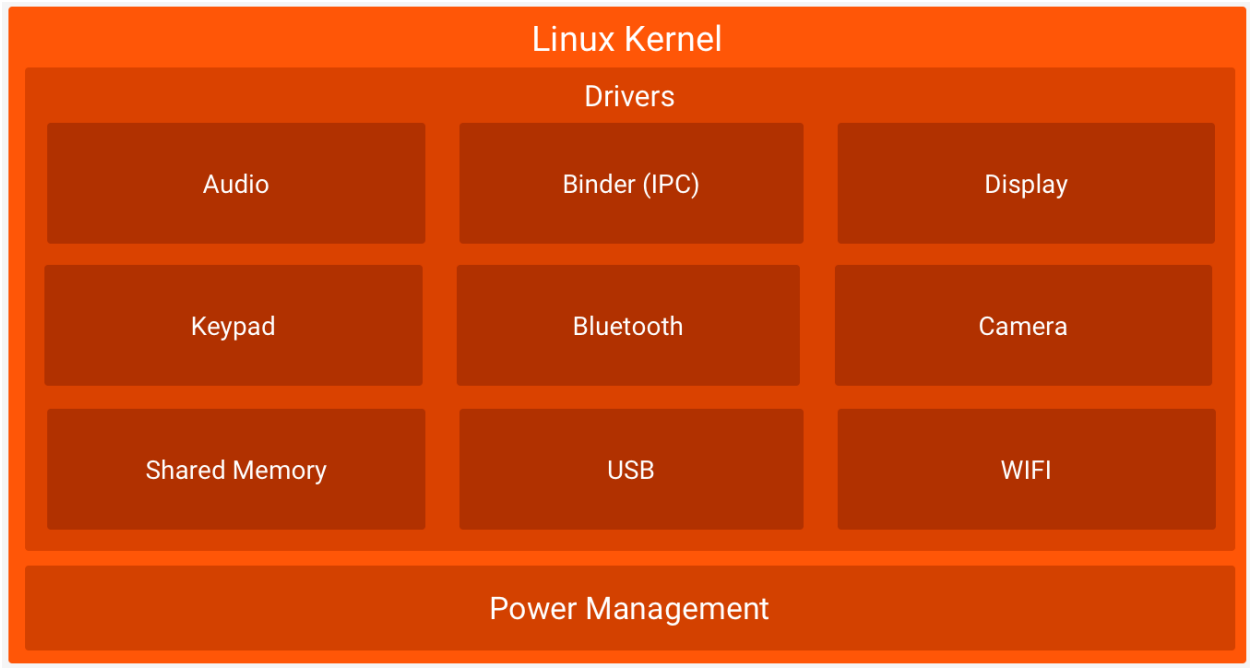

Figure 17: Linux Kernel

It is the foundation of the Android platform which lies the bottom of the layers. Using a Linux kernel allows Android to take advantage of crucial security features and enables device manufacturers to develop hardware drivers for a well-known core (Platform Architecture, 2018). Linux kernel is responsible for device drivers, power management, memory management, device management and resource access (Android Software Stack, 2018).

Hardware Abstraction Layer (HAL)

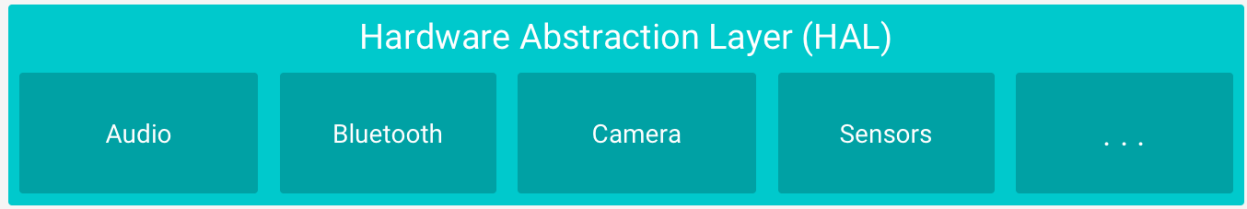

Figure 18: Hardware Abstraction Layer (HAL)

HAL provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware components, such as the camera or Bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component (Platform Architecture, 2018).

Android's HAL uses the functions provided by the lower-layer Linux kernel to serve the request from the Android application/framework. The HAL implementation is hardware-specific and varies from vendor to vendor. Developers need a standardized approach to implement the HAL across vendors and hardware to reduce development time, cost and effort. HAL also allows developers to perform functions without affecting or modifying the higher-level system (Sarkar, 2017).

Android Runtime

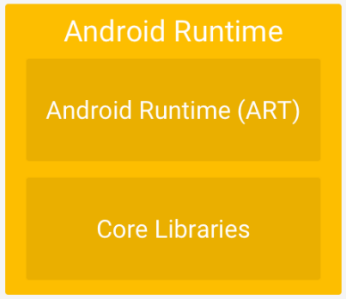

Figure 19: Android Runtime

For devices running Android version 5.0 or higher, each app runs in its process and with its instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed especially for Android optimized for minimal memory footprint (Platform Architecture, 2018).

ART is a runtime from Android 5.0 (Lollipop), and it has completely replaced Dalvik Virtual Machine (DVM). Android 7.0 adds a just-in-time (JIT) compiler which was used in Dalvik with code profiling to Android runtime (ART) that continually improves the performance of Android apps as they run (Sinhal, 2017).

Native C/C++ Libraries

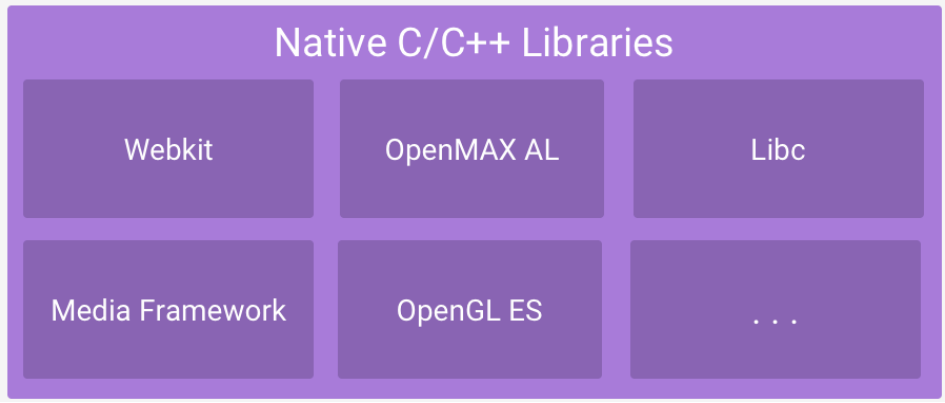

Figure 20: Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that requires native libraries written in C and C++ programming language. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. For example, it is possible to access OpenGL ES through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in the app (Platform, 2018). A summary of some essential core Android libraries available to the Android developer is as follows:

- Android. App − Provides access to the application model and is the cornerstone of all Android applications.
- Android. Content − Facilitates content access, publishing and messaging between applications and application components.
- Android. Database − Used to access data published by content providers and includes SQLite database management classes.
- Android. OpenGL − A Java interface to the OpenGL ES 3D graphics rendering API.
- Android. Os − Provides applications with access to standard operating system services including messages, system services, and inter-process communication.
- Android. Text − Used to render and manipulate text on a device display.
- Android. View − The fundamental building blocks of application user interfaces.
- Android. Widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, and radio buttons
- Android. Webkit − Applications that use a set of classes intended to allow web-browsing capabilities

Source: Android Architecture, 2018

Java API Framework

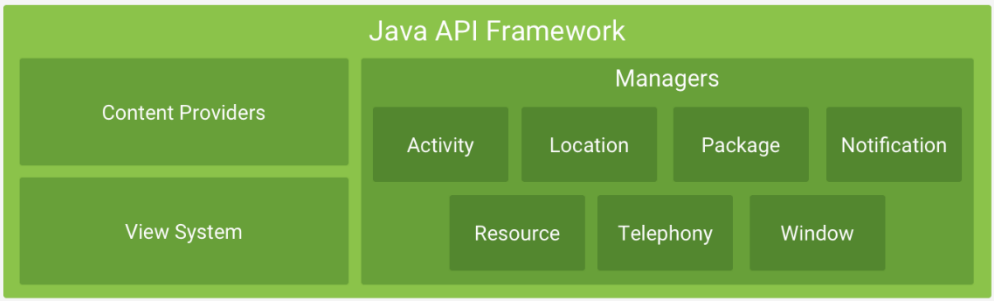

Figure 21: Java API Framework

The entire feature-set of the Android OS is available to developers through APIs are written in the Java language. Application developers can make use of these services in their applications. These APIs form the building blocks needed to create Android apps by simplifying the reuse of core, modular system components, and services (Platform, 2018). It consists of tools for designing UIs like buttons, text fields, image panes, and system tools like intents, phone controls and media players (Murdock, Casey, Shaji, & Rishi, n.d.).

The Android framework includes the following vital services:

- Activity Manager − Controls all aspects of the application lifecycle and activity stack.
- Content Providers − Allows applications to publish and share data with other applications.
- Resource Manager − Provides access to non-code embedded resources such as strings, color settings and, user interface layouts.
- Notifications Manager − Allows applications to display alerts and notifications to the user.
- View System − An extensible set of views used to create application user interfaces.

Source: Android Architecture, 2018

System Apps

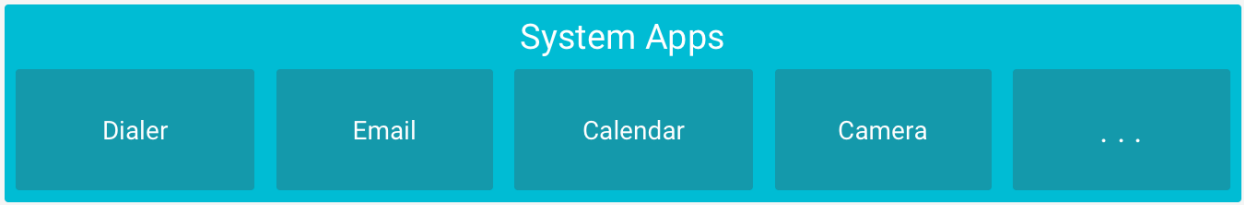

Figure 22: System Apps

Android comes with a set of core apps for email, short message service (SMS) messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So a third-party app can become the user's default web browser, SMS messenger, or even the default keyboard depending on the system's Settings app. The system apps function both as apps for users and to provide critical capabilities that developers can access from their app. For example, if the user would like to deliver an SMS, the user does not need to build that functionality but can instead invoke whichever SMS app is already installed to provide a message to the recipient (Android Architecture, 2018).

Literature Related to the Problem

The sources cited in this paper were not from scholarly journals but on online websites. Although less credible than peer-reviewed, the references were cross-checked to determine its accuracy. The sources cited accumulated were from these main groups of related sources:

- Antivirus websites - No one has a better understanding of malware than antivirus companies such as Norton, Avast, AVG, Comodo, and Malwarebytes. Initially labeled as antivirus companies due to viruses being a real threat in the early days of computing have grown to encompass all threats on different platforms. Companies that produce protective software are on the front lines of malicious attacks. Their software is continually being updated with new virus and malware definitions to counter further attacks which make the documentation on their sites the most recent and accurate.

- Security blogs/ sites – These are referring to reputable companies in the industry such as Cisco and Heimdall security. The experts provide reliable and sound advice to any issues facing their consumers
- Wikipedia – Not necessarily part of the group of sources, it was used to provide definitions for the keywords on the topic. Wikipedia's content is editable by the public which makes it unreliable for information use.
- Android websites – Research on sites like Android Authority and Android Pit as it contained up to date information in all things related to Android. Android's official developers' website was especially helpful in describing the architecture
- Android Books – Almost most books researched on Android had a section on security. Credible and peer-reviewed, it lacked in terms of the most recent security updates. For this reason, websites/ blogs were the citations for most of the sources. It is for the paper to be up to date for users and developers of the platform.

Literature Related to the Methodology

Krajci & Cummings (2013) explained the overview, history, and evolution of the Android platform and contrast with other competitors specifically the Apple iPhone. It also focuses on application and platform security

Andrews, Oh, and Stackpole (2013) perform focused research on finding better methods of malware analysis that will ensure the development of better malware defensive measures. They developed a necessary environment for an educational organization to introduce Android-based malware research in a lab format.

Yuhui and Ning (2014) use static and dynamic behavioral analysis approach to analyze Android malware. The research paper yields its malicious behaviors and its ways of stealing private data and provides methods of detection and prevention.

Rasthofer (2017) proposed a reverse engineering framework that included different approaches for automatically extracting insights into the behavior of an Android application. His approach combines static and dynamic code analysis techniques based on code fuzzing in such a way that it is resistant against common obfuscation techniques. The methods provided different insights into the analyzed application, in particular, how and under which circumstances the application communicates with its environment.

Summary

This chapter described the Android architecture as well as current detection methods being used to detect malware. The literature reviewed pertains to detecting malware using static and dynamic analysis as well as analyzing malware in a controlled environment. Additional research approaches and methods are required to gain more insight into malware behavior.

Chapter III: Methodology

The framework of the study uses the qualitative approach. Qualitative research is used to survey the commonality of bugs and types of malware attacks affecting the platform. Due to this research method, the paper presents the most common attacks currently affecting Android devices.

The qualitative research methods used were reviews and observation (Karg, 2012). Reports that include combing through scholarly literature, published writings and reputable websites to determine the methods used to exploit the device's vulnerability.

Data Collection

In qualitative research, obtaining the history of the Android platform from websites, research papers, and YouTube was a form of data collection. Analysis obtained was used to install custom OS developed by open source developers on Android devices and determined that not every Android device has the most current version which makes it susceptible to attacks.

Installing a custom OS proved that average users with minimal advanced knowledge using instructions and software available online could easily manipulate Android devices. It is possible that an average user is only able to install 3rd party applications by circumnavigating the official Google Play Store which supplies apps that have been vetted and deemed safe to use. Google and device manufacturers have countered this by hiding the option for allowing installation for 3rd party applications. It is unknown if this effort is intentional as, through observations of several Android devices (smartphones and tablets), the official release of the OS has been modified by the manufacturer to adhere to their hardware and user interface specifications. The option to use 3rd party software was especially challenging to find in the Samsung Galaxy S8.

Data Analysis

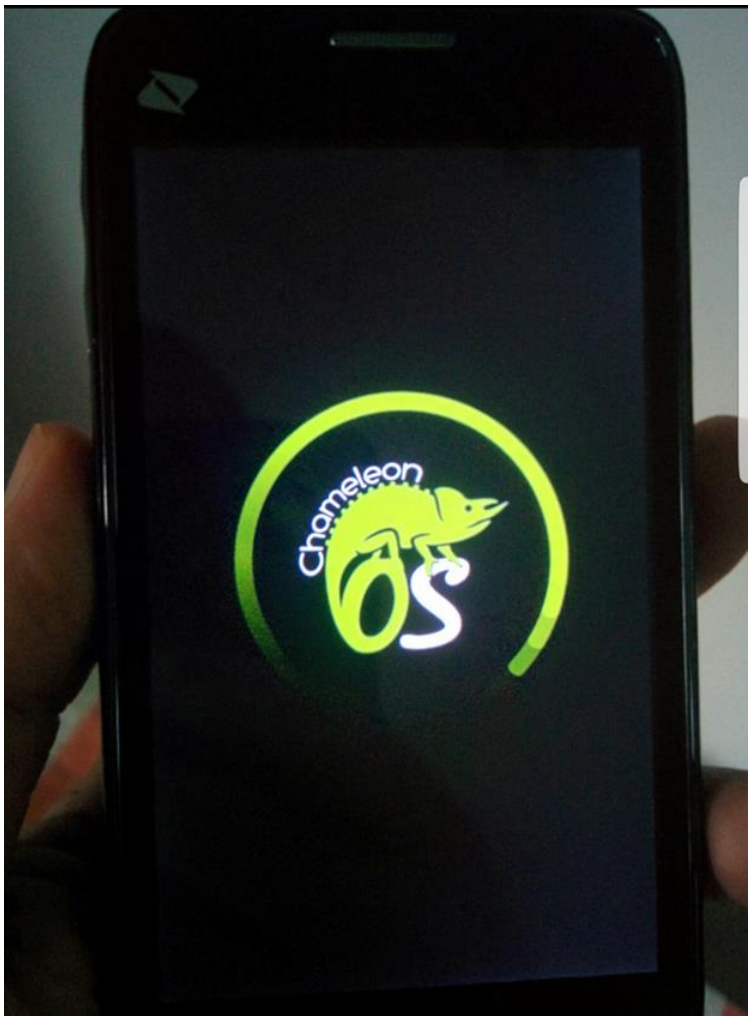

Figure 23: Custom Chameleon firmware (CFW) installed on ZTE Warp N860

The tools used to analyze the data was using a laptop, third-party software obtained online with an internet browser for research. Also used was two Google Nexus 7 tablets, a Samsung Galaxy S3 phone, a Google Nexus 5 phone (Figure 24), and a ZTE Warp N860 phone (Figure 23). Research on specific websites was crucial as there were conflicting data from sites seeking to minimize the risks to sell Android devices.

The installation of Custom Firmware (CFW) enabled better understanding for the Android architecture as mentioned in 'Data Collection'. Research on security websites and blogs as well as scholarly journals allowed a better understanding of the loopholes present in the present and past Android versions.

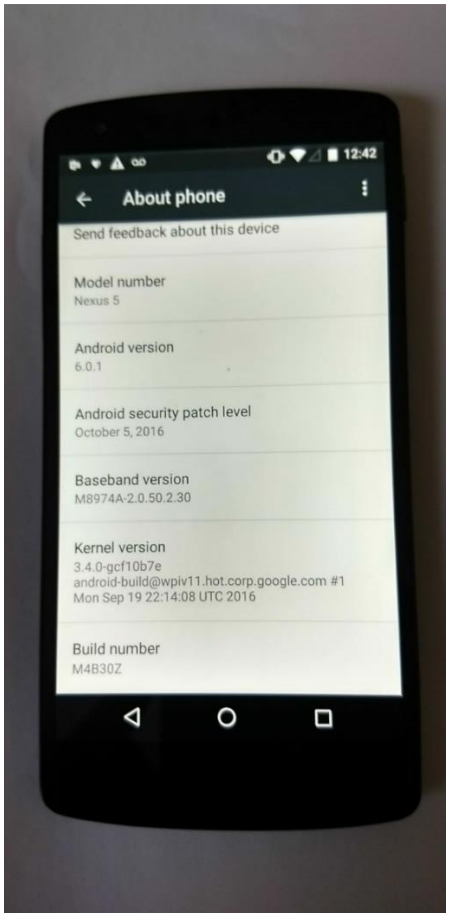

Figure 24: Google Nexus 5 phone with original firmware (OFW)

Installation of the CFW has both its pros and cons. Google supplies updates and security patches for only two years after the device release. After this period, the device is susceptible to attacks and malware. It is for this reason that the user resorts to CFW. CFW might contain the updated OS version as well as the security updates, but the user is at the mercy of these open source developers as there could be malicious intent and potential bugs. From observation and

research of the forums and community contributing to these developments, CFW is vetted and tested. They provide reviews of the performance and bugs encountered to facilitate the improvement of the CFW further. The more popular the device, the better chance it has to be updated. One of these devices is the Samsung Galaxy S3 released in 2012 had sold about 70 million units by 2015. It was so popular that Samsung released a refreshed version called the "Galaxy S3 Neo" in April 2014 which had an updated CPU and RAM (Samsung Galaxy S III, n.d.). Six years after its release and far from receiving support from Google, it remains usable due to the efforts of the open source community.

Summary

This chapter covered the framework of the study using the qualitative approach. The useful tools and techniques described in this chapter determined the types of attacks described in the next section.

Chapter IV: Prevention

The purpose of the previous sections (chapters) is to educate the reader on the background of the Android OS so to better understand its vulnerability to attacks. As noted, not all attacks incurred on the Android platform is the fault of the platform itself in terms of bugs but is mostly be attributed mostly to the user. Social engineering, in the context of information security, refers to psychological manipulation of people into performing actions or divulging confidential information (Social engineering (security), n.d.). A bank can have the most secure vault, but it is in vain if the key to secure it falls in the wrong hands. As much as developers try to defend the platform, it will never be indeed 100% secure.

In this section, prevention and mitigation techniques will be suggested and proposed to counter these attacks. Although personal computers and other mobile platforms share common methods to discourage malicious attacks, this section focuses on the Android platform.

Addressing Android Bugs

Wi-Fi, Bluetooth Mobile data, GPS

When it comes to system bugs, there is no prevention as the bug already exist. While it is up to the developers to fix it and issue patches to the operating system, users have the only option of waiting for the software patch update or purchasing a new device. Users, however, do have other options which might not be permanent but will delay the purchase of a new phone or tablet. These steps listed below have been recorded in forums by users of similar devices who have provided temporary solutions to these issues. Here are the steps to be followed should a user encounters a problem with their equipment:

1. Restart
2. Safe mode

3. Wipe cache

4. Backup

5. Factory Reset

Restart device. Restarting the phone clears open apps and memory leaks and gets rid of anything draining the user's battery. Crashing happens for various reasons, but not rebooting the phone can have an effect here as well, as every update, and an app installed or deleted will add and remove the code in the operating system of the phone. Restarting the device will eliminate most of these issues and, it will work better (Walansky, 2017).

Restarting a phone in normal conditions working is typically a matter of minutes. If the device is frozen, there is a way the user can force reboot the phone. Press and hold down the power button along with the volume up button until the screen goes off. Power the device back on pressing the power button for a few seconds (How to Restart Your Android Phone?, n.d.)

Safe Mode. Safe mode is a way to launch Android on a smartphone or tablet without any third-party apps that might ordinarily run as soon as the operating system finishes loading. Usually, when powering on an Android device, it may load a series of apps automatically like a clock or calendar widget on the home screen. Safe mode prevents this from happening, which is great if the Android smartphone or tablet is crashing frequently or running incredibly slow. However, it is a troubleshooting tool rather than an actual cure for the problem. When the user launches an Android smartphone or tablet in safe mode, third-party apps cannot run at all even after the device boots up (Nations, 2018).

Booting the device in Safe Mode narrows down what might be causing it to crash or to run abnormally slow. If the smartphone or tablet runs fine in safe mode, it is not the hardware

causing the problem. If it is not a hardware issue, the user needs to figure out which app is causing the problem (Nations, 2018).

To boot into Safe Mode, follow the following steps. Press and hold the Power button, then tap and hold on the Power off option, and then tap OK to boot into safe mode. If the problem is gone, then a third-party app installed is the cause. It can be either uninstall one-by-one and retest or go for a factory reset and install apps selectively (Hill, 2016).

Wipe cache partition. The system cache partition stores temporary system data. Cache helps the system to access the apps and its data more quickly, but sometimes it's getting outdated. Cache cleaning is a process which is done in a certain interval of time to help the system run smoothly. Cache cleaning is different from the factory reset so it will not affect personal or internal data. It is also recommended to have a cache cleaning after a system update as well (How to Wipe Cache Partition on Android?, n.d.). Use the following steps to wipe the cache partition:

1. Turn off the device.
2. Hold down the Power button and the Volume down button until the Android mascot is on his back.
3. Use Volume down to highlight Recovery mode and Power to select it.
4. Press and hold Power and Volume up for three seconds, then let go of Volume up, but keep holding Power.
5. When the options menu is visible, the user can use the volume keys to highlight wipe cache partition and the Power key to select it.

Source: Hill, 2016

Backup. The purpose is to create a copy of data in the event of a data failure. Storing it on a separate medium is critical to protecting against data loss or corruption (What is Backup and Recovery?, n.d.).

Google's Android offers the ability to seamlessly save specific settings like wireless network preferences, bookmarks, and custom dictionary words to their servers using your Google account. The following steps will enable it:

1. Go to Settings, Personal, Backup and reset, and select both Back-ups my data and Automatic restore.
2. Go to Settings, Personal, Accounts & Sync, and select your Google account.
3. Select all the option boxes listed.

Source: King, 2012

To back up pictures, videos and essential data manually, the user can back up their device to the computer manually by following the steps below:

1. Connect the device to the PC via a USB cable, and it will show up as an external hard drive.
2. Select the disk and navigate to the DCIM folder. This folder contains your video and picture data.
3. Select the data files to back up, and drag them to an area on the computer, such as the desktop. It will copy over to your computer.

Source: King, 2012

Factory Reset. Factory Reset will restore the Android device to the state where it was made out in the factory. It implies that all installed applications, software, passwords, accounts

and other personal data that the user may have stored on the internal phone memory will be wiped out clean. Below are two ways to factory reset a device:

- How to Factory Reset a phone - The easiest way to factory reset a phone is through the settings menu. The exact location of the factory reset option varies based on the phone, but once the user finds the backup and reset menu, they will see a Factory Data Reset button which the user should tap to set it off.
- Factory Reset in Recovery Mode - The procedure for recovery mode factory reset is not the same for every smartphone. So, the user may take the other initiatives in finding out the exact steps for their model. But it usually involves holding the volume down key/volume up key and Power key together. Use the volume keys to choose wipe cache partition. Once done, select wipe data/factory reset to delete any settings or apps that could be causing the malfunction. From there, the user can reboot the phone to see if the problem still exists.

Source: Albert, 2018

Other Android bugs

Camera not working. There are other potential reasons for the camera to crash or return an error message and refuse to load. Here are steps to resolve it:

1. Go into Settings > Apps > Camera and tap Force stop
2. Tap on Clear cache and Clear data.
3. Hold down the Power button and select Restart.

If there is another app installed that is accessing the camera, then it could be the issue. The user can test this by booting into Safe Mode (refer to ‘Safe mode’ section under ‘Addressing

Bugs'). If the problem is gone, then the user can surmise that a third-party app is a cause. Try uninstalling the apps that use the camera to find the culprit. Try wiping the cache partition (refer to 'Wipe Cache Partition' section under 'Addressing Bugs').

Last option would be a factory reset. If there are still issues after a factory reset, then it could be a hardware fault. On some phones pressing gently around the camera sensor can help restore a loose connection. If that is the case, then the user would probably want to contact the manufacturer or the carrier and find out about a repair or replacement (Hill, 2016).

Overheating. Android 4.4.4 (KitKat) users have reported overheating, even when the phone is not in use and doesn't seem to be doing anything. The most likely cause for this is a third-party application, such as Facebook, Viber, Skype and such. The first thing the user needs to do is identify whether a third-party app is indeed causing the overheating problem by waking the phone up too often or soliciting it a lot.

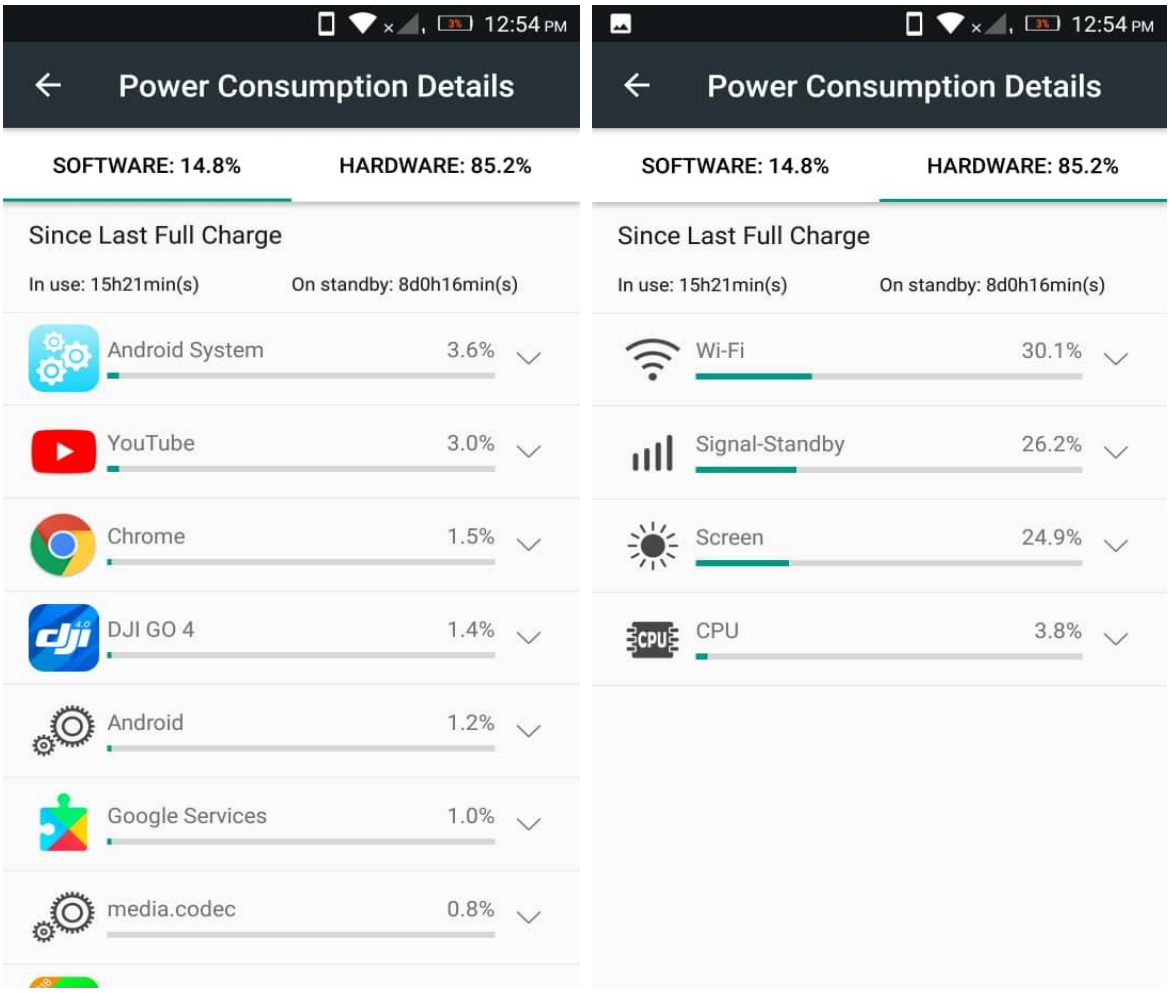

Figure 25: Power Consumption Details

The user can do that by starting the phone in Safe Mode (refer to ‘Safe mode’ section under ‘Addressing Bugs’) and monitoring it for a few hours, to see if the battery is draining the same way. If it’s not draining, then the problem is with a third-party application. Finding out which of the apps is causing the battery drain is a little trickier, but usually, the ‘Power Consumption Details’ in Figure 25 shows which apps are using most of the battery power, and the one on top of the list is typically the culprit. It is better to delete that app (Kilin, 2015).

Overheating is often a side-effect of really hammering the phone with apps such as 3D games. The device casing might also be contributing to the problem as well – phones are designed to be case-free, and bulky cases can interfere with heat dissipation (Carlon, 2017).

Performance problems. Installing updates on top of updates on top of updates can eventually cause problems. A factory reset might be beneficial, but it is best to see if the current OS is up to date. As with any performance issues, the user should try switching off anything not need to know if they can identify a specific cause (Carlon, 2017).

Just turn the phone off for a minute before restarting it might fix the issue. The next step is to try to identify the problem, to see whether it's app-related or the system itself. It is best to boot into safe mode (refer to Safe Mode under Addressing Bugs). Safe mode disables all apps installed so the user can assess whether the core system apps are functioning normally. If they are, then one of the installed apps is the problem, and the user should try systematically uninstalling them one at a time to find the culprit.

If the Android device is unstable even in safe mode, then it's an OS problem, not an app problem. The user can try clearing the system cache by following the steps outlined in the 'Wipe Cache Partition' under Addressing Bugs. If that still hasn't fixed the problem, the last resort is to back up the data and factory reset (refer to 'Factory Reset' under Addressing Bugs) the phone.

Addressing Android Malware

Downloading software and files. It is best to think twice before immediately downloading and installing any new software—especially freeware (Adware, n.d.). Download programs only from reputable sources because threat actors have been known to package and distribute malware as security software (Rouse, 2018). Users should be wary of installing apps

from untrusted sources such as private markets and messages or websites offering free apps for installation. When installing a new app, it is best to check the list of permissions to see if it is appropriate for the app the user is installing (Savage, Coogan, & Lau, 2015). Users should avoid torrent sites and downloading files from unknown sources and dubious websites.

Reputable cybersecurity software. Due to its popularity and most widely-used operating system, having security apps and antivirus tools installed on the user's device is essential. Besides running automatic scans, the software will actively try to prevent malicious web pages and files from being opened or downloaded (Drake, 2019). Users should download a popular cybersecurity program for their device as well as perform periodic diagnostic scans. Users should always make sure to keep the cybersecurity software and the Operating System (OS) update-to-date.

Firewalls. In computing, a firewall is a software or firmware that enforces a set of rules about what data packets will be allowed to enter or leave a network. Android uses Application Sandboxing and the Permissions Model to act as a firewall. The goal of sandboxing is to improve security by isolating an application to prevent outside malware, intruders, system resources or other apps from interacting with the protected app (Rouse, 2012). In the Permissions Model, the Android system installs every Android application with a unique user and group ID. Each application file is private to this generated user. Therefore, utilizing the underlying Linux kernel, every Android application is isolated from other running applications. Below are apps that function as a firewall currently available:

- DroidWall - requires root access and allows users to select which apps can and cannot have Internet interaction.

- NoRoot Firewall - This app enables firewall without the user having root permissions. To use this, users must install the app and then tick boxes off to allow apps and features Internet access and deny access to others as desired.

Source: Jones, 2017

These apps pose an added security risk. Not only can the app itself be used to access personal information but if a user is unaware of the intentions of a program, it could be malicious. The DroidWall app requires root access which would mean bypassing security protocols put in place by the Android OS.

Passwords. It would help the user to create a unique password for each account using a complex combination of letters, numbers, and symbols. Most passwords are attainable through social engineering and can be as simple of an act like writing the password on a post-it note. With the Google Account password, the attacker can access Google account details such as Gmail, Google drive, credit card information used on the Google Play Store. There are multiple security features available to the user to unlock the device and gain access to information and functions. The following are the unlock techniques:

- Phone Security
    - Swipe - Swipe a finger across the screen. It gives no protection, but to get to the Home screen quickly.
    - Pattern – This is a standard Android pattern unlock method, whereby the user traces a line the user sets through a grid of nine dots.
    - PIN / Password - The user can use a four-digit PIN code or alphanumeric password to unlock the device.

Source: Set screen lock on an Android device, n.d.

- Biometrics
    - Face Recognition - This method uses the front-facing camera to identify the user and unlock the device. Some users found it problematic to use even in good or poor light, and when it did work, it would take several seconds of awkwardly staring at the phone before it would unlock.
    - Iris Scanner - Samsung has included an iris scanning method, which uses sensors on the front of the phone to identify the user and unlock the device. Using the iris does have its downsides: though it works in low light, it can have a lot of trouble recognizing the eyes in sunlight, and the user has to hold the phone up and awkwardly close to their face.
    - Fingerprint Scanner - Unlike other devices, such as the Google Pixel that put the scanner in the middle of the phone where the index finger naturally rests, the Samsung Galaxy S8's scanner is high up on the back of the device and not in the middle.

Source: Seifert, 2017

Android Device Manager is a security feature that helps users locate, and if needed, remotely lock, unlock or wipe the Android device if users happen to lose it or it gets stolen. Device Manager works to protect the Android device. All the user needs to do is connect the device with your Google account (Android Device Manager, 2018).

Summary

This chapter covered prevention, mitigation as well as steps that can be taken by user and developers alike should they encounter issues. Android users do not have a dedicated Apple Store where they could go to get their devices fixed. Instead, they must rely on the open source community and limited manufacturer support to help them resolve these issues. Although issues and attacks may seem as separate categories, the techniques to fix it are similar. This chapter is the accumulation of the teamwork between developers and users alike to resolve issues and counter attacks on the Android platform.

V. Conclusion

Society's reliance on mobile devices has made it impossible to ignore the security implications that pose its users. Despite the assurance that the Android platform will protect its users due to its implemented security measures, attackers will always find a way to circumvent these measures. This research serves as a guideline to educate all users to defend themselves and in the worst-case scenario, if infected; what measures to take to remedy the situation. Most users take security for granted as they assume it does not play a role. The research in this document is meant to raise awareness among users with that mindset. The limitations of this research are mostly attributable to social engineering which can circumvent all preventive measures presented. Measures posed in this paper should be applicable in future Android versions unless there are significant changes in the architecture.

These changes are welcome as attackers will have to revamp their strategy to cope with the new changes. Advancements in technology mean more modern hardware and software to match. It means faster processing and increased battery life. Monitoring forums for user feedback would be the best way to determine if the issue is a bug or an attack. Quantum computing is inevitably the future of computing. This technology is still at its infancy, and it will be a slow and gradual change, but eventually, all devices will encompass this new technology. Attacks on current computing devices (traditional computing) will not affect quantum computing until this new technology has matured.

Further protective measures will surpass passwords and biometrics. With advancing technology, more capable sensors will be added to mobile devices making it a reality. Modular smartphones will use these sensors to enable users to swap modules. As we can only prepare for

the future, the cooperation of users and developers would ensure some peace of mind for users of this platform.

References

Adware - What Is It & How To Remove It. (n.d.). Retrieved June 27, 2018, from https://www.malwarebytes.com/adware/

Albert, K. D. (2018, February 02). *When, why and how you should factory reset your Android smartphone*. Retrieved March 12, 2018, from https://www.dignited.com/28006/should-you-factory-reset-your-android-smartphone/

An Overview of the Android Architecture. (n.d.). Retrieved October 20, 2017, from http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture#The_Android_Software_Stack

Andrews, B., Oh, T., & Stackpole, W. (2013, June). Android malware analysis platform. *In Proc. of 8th Annual Symposium on Information Assurance (ASIA'13).*

Android Architecture. (n.d.). Retrieved September 27, 2018, from https://www.tutorialspoint.com/android/android_architecture.htm

Android Device Manager. (2018, November 05). Retrieved January 16, 2019, from https://www.itarian.com/android-device-manager.php

Android Software Stack. (n.d.). Retrieved October 30, 2018, from https://www.javatpoint.com/android-software-stack

Anti-Phishing Working Group (APWG), (2006a). *Phishing activity trends report*, available online at: http://www.antiphishing.org/reports/apwg_report_april_2007.pdf (last accessed 14 December 2007).

Brewster, T. (2018, January 16). *One of the 'most powerful' Android spyware tools ever was just uncovered.* Retrieved October 23, 2018, from https://www.forbes.com/sites/

thomasbrewster/2018/01/16/android-spyware-found-by-kaspersky-negg-italyl/#63ffed111979

Capture and read bug reports. (n.d.). Retrieved October 4, 2018, from https://developer.android.com/studio/debug/bug-report

Carlon, K. (2017, January 23). *Disastrous Android Lollipop problems and their solutions.* Retrieved September 28, 2018, from https://www.androidpit.com/android-5-0-lollipop-problems-and-solutions

Cluley, G. (2012, September 13). *Android rootkits – malware on your smartphone.* Retrieved November 14, 2018, from https://nakedsecurity.sophos.com/2010/06/02/android-rootkits-malware-smartphone/

Correspondent, N. (2014, April 16*). Google patches Android flaw that allows phishing apps to spoof genuine ones.* Retrieved August 30, 2018, from https://gadgets.ndtv.com/mobiles/news/google-patches-android-flaw-that-allows-phishing-apps-to-spoof-genuine-ones-509316

Cucu, P. (2019, January 29). *Rootkit - the (nearly) undetectable malware*. Retrieved July 25, 2018, from https://heimdalsecurity.com/blog/rootkit/

Dibben, C. (2009, October 26). *Donut (Android 1.6) glitches/problems*. Retrieved September 19, 2018, from https://forum.vodafone.co.uk/t5/Archive/Donut-android-1-6-Glitches-problems/m-p/264761

Donohue, B. (2014, April 23). *SMS Android Trojan targets users in United States*. Retrieved May 12, 2018, from https://www.kaspersky.com/blog/fakeinst-targets-us-users/4601/

Drake, N. (2019, January 03). *The best Android antivirus 2019*. Retrieved January 27, 2019, from https://www.techradar.com/sg/news/the-best-antivirus-for-android-in-2018

DuPaul, N. (2017, July 21). *Common malware types: Cybersecurity 101*. Retrieved October 20, 2017, from https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101

Fruhlinger, J. (2018, December 19). *What is ransomware? How these attacks work & how to recover from them.* Retrieved January 13, 2019, from https://www.csoonline.com/article/3236183/ransomware/what-is-ransomware-how-it-works-and-how-to-remove-it.html

How to Restart Your Android Phone? (n.d.). Retrieved May 30, 2018, from https://drfone.wondershare.com/reset-android/restart-android-phone.html

How to Wipe Cache Partition on Android? (n.d.). Retrieved February 2, 2019, from https://drfone.wondershare.com/erase-android/wipe-cache-partition.html

Hill, S. (2016, February 9). *Android 5.0 Lollipop: Common problems and how to fix them | Page 3.* Retrieved September 4, 2018, from https://www.digitaltrends.com/mobile/android-lollipop-problems/3/

Jareth. (2018, September 11). *What is a computer worm and how does it spread?* Retrieved December 12, 2018, from https://blog.emsisoft.com/en/28154/computer-worms/

Jones, B. (2017, November 22). *How to enable a firewall on your Android*. Retrieved August 13, 2018, from https://www.psafe.com/en/blog/how-to-enable-a-firewall-on-your-android/

Judge, K. (2018, August 3). *What is a computer virus? | Types of computer viruses* [Updated 2019]. Retrieved October 17, 2018, from https://antivirus.comodo.com/blog/computer-safety/what-is-virus-and-its-definition/

Karg, C. (2012, October 14). *Qualitative vs. quantitative research*. Retrieved November 03, 2017, from https://www.aiuniv.edu/blog/2012/october/qualitative-vs-quantitative-research

Kilin, E. (2015, March 11). *Android 4.4.4 KitKat bugs and how to fix them*. Retrieved May 12, 2018, from http://www.loadthegame.com/2015/03/11/android-4-4-4-kitkat-bugs-fix/

King, M. (2012, February 03). *How to back up your Android phone.* Retrieved November 16, 2018, from https://www.pcworld.com/article/248984/how_to_back_up_your_android_phone.html

Krajci, I., & Cummings, D. (2013). History and evolution of the Android OS. In *Android on x86* (pp. 1-8). Apress, Berkeley, CA. (2013 Android book)

Life of a Bug. (n.d.). Retrieved December 12, 2018, from https://source.android.com/setup/contribute/life-of-a-bug

Lipovsky, R. L. S., & Branisa, G. (2015). The rise of Android Ransomware. *Enjoy safer technology.* Retrieved July 5, 2017, from https://www.welivesecurity.com/wp-content/uploads/2016/02/Rise_of_Android_Ransomware.pdf

Marshall, C. (2016, June 06). *Android Marshmallow problems and how to fix them*. Retrieved March 16, 2018, from https://www.androidpit.com/android-marshmallow-problems-and-how-to-fix-them

McGhee, B. (2017, July 2). Android Nougat problems and their solutions. Retrieved October 16, 2018, from https://www.androidpit.com/android-nougat-problems-solutions#installation

Milin-Ashmore, J. (2016, September 26). *5 of the Most Dangerous Android Viruses and How to Get Rid of Them*. Retrieved June 3, 2018, from https://www.maketecheasier.com/dangerous-android-viruses/

Murdock, M. T., Casey, B., Aziz, S. (Shaji), & Rishi. (n.d.). *Android framework. What is it?* Retrieved May 27, 2018, from https://stackoverflow.com/questions/2968016/android-framework-what-is-it

More Than 1 Million Google Accounts Breached by Gooligan. (2017, March 29). Retrieved December 7, 2018, from https://blog.checkpoint.com/2016/11/30/1-million-google-accounts-breached-gooligan/

Narmatha, M., & KrishnaKumar, S. V. (2016). Study on Android operating system and its versions. *International Journal of Scientific Engineering and Applied* Science *(IJSEAS),* 2(2), 439-444.

Nations, D. (2018, December 22). *How to get out of Android's safe mode.* Retrieved January 30, 2018, from https://www.lifewire.com/android-safe-mode-4158035

O'Donnell, A. (2018, September 12). 5 types of malicious bots and how to avoid them. Retrieved December 25, 2018, from https://www.lifewire.com/what-are-malicious-bots-2487156

Platform Architecture. (2018, October 11). Retrieved December 5, 2018, from https://developer.android.com/guide/platform/

Ransomware - What is it & how to remove it. (n.d.). Retrieved September 5, 2018, from https://www.malwarebytes.com/ransomware/

Rasthofer, S. (2017). *Improving mobile-malware investigations with static and dynamic code analysis techniques.* (Doctoral dissertation), Technische Universität Darmstadt.

Raymond, J. (2018, September 01). *What is computer worm? | How this virus spreads and infects PC.* Retrieved November 28, 2018, from https://antivirus.comodo.com/blog/comodo-news/computer-worm-virus/

Rouse, M. (2012, November). *What is application sandboxing? - Definition from WhatIs.com.* Retrieved September 16, 2018, from https://searchmobilecomputing.techtarget.com/definition/application-sandboxing

Rouse, M. (2018, January). *What is Trojan horse (computing)? - Definition from WhatIs.com*. Retrieved June 30, 2018, from https://searchsecurity.techtarget.com/definition/Trojan-horse

Rouse, M. (2018, April). *What is rootkit? - Definition from WhatIs.com*. Retrieved December 12, 2018, from https://searchsecurity.techtarget.com/definition/rootkit

Sarkar, P. (2017, November 21). *Android hardware abstraction layer implementation - The challenges and way around - Aricent Altran group blog.* Retrieved July 5, 2018, from https://connect.aricent.com/2017/07/android-hardware-abstraction-layer-implementation-the-challenges-and-way-around/

Savage, K., Coogan, P., & Lau, H. (2015). The evolution of ransomware. *Symantec blog*. Retrieved August 4, 2017.

Set screen lock on an Android device. (n.d.). Retrieved from https://support.google.com/android/answer/9079129?hl=en

Shah, V. (2016, August). (PDF) *Android operating system revolution in mobile technology*. Retrieved February 23, 2018, from https://www.researchgate.net/psaublication/306358743_Android_operating_system_revolution_in_mobile_technology

Shinde, G. (2017, March 28). *New Android ransomware bypasses all antivirus programs*. Retrieved November 28, 2018, from https://www.zscaler.com/blogs/research/new-android-ransomware-bypasses-all-antivirus-programs

Shipman, M. (2011, January 28). *Data leak vulnerability haunts latest Android (Gingerbread).* Retrieved July 24, 2018, from https://news.ncsu.edu/2011/01/haunted-android/)

Sims, G. (2017, August 24). *How Oreo is better than Nougat: Introduction*. Retrieved May 5, 2018, from https://www.androidauthority.com/android-oreo-vs-android-nougat-introduction-794696/

Sinhal, A. (2017, April 05). *Closer look at Android Runtime: DVM vs. ART – AndroidPub.* Retrieved July 30, 2018, from https://android.jlelse.eu/closer-look-at-android-runtime-dvm-vs-art-1dc5240c3924

Spyware. (n.d.). Spyware - What is it & how to remove it. Retrieved January 10, 2018, from https://www.malwarebytes.com/spyware/

Srivastav, S. (2018, October 08). *Android P 9.0 vs. Android O 8.0 - The War of the OS.* Retrieved June 5, 2018, from https://appinventiv.com/blog/android-9-pie-vs-android-8-oreo

Torres, G. (2017, December 18). *What is a computer virus? | The ultimate guide to PC viruses*. Retrieved June 12, 2018, from https://www.avg.com/en/signal/what-is-a-computer-virus

Walansky, A. (2017, February 16). *Is it bad to never restart your phone?* Retrieved July 28, 2018, from https://www.rd.com/culture/restarting-your-phone/

What are bots? (n.d.). Retrieved August 28, 2018, from https://us.norton.com/internetsecurity-malware-what-are-bots.html

What is a computer virus? (n.d.). Retrieved February 24, 2018, from https://us.norton.com/internetsecurity-malware-what-is-a-computer-virus.html

What is a computer worm and how does it work? (n.d.). Retrieved June 30, 2018, from https://us.norton.com/internetsecurity-malware-what-is-a-computer-worm.html

What is a Trojan? Is it a virus or is it malware? (n.d.). Retrieved October 25, 2018, from https://us.norton.com/internetsecurity-malware-what-is-a-trojan.html

What is Adware? (n.d.). Retrieved September 28, 2018, from https://www.kaspersky.com/resource-center/threats/adware

What is Backup and Recovery? (n.d.). Retrieved October 28, 2018, from https://www.netapp.com/us/info/what-is-backup-and-recovery.aspx

What is malware? A guide to the 12 most common types. (2018, December 04). Retrieved January 27, 2019, from https://www.quostar.com/blog/what-is-malware/

What is phishing? (n.d.). Retrieved March 7, 2018, from http://www.phishing.org/what-is-phishing

What is spyware? And how to remove it. (n.d.). Retrieved August 12, 2018, from https://us.norton.com/internetsecurity-how-to-catch-spyware-before-it-snags-you.html

What is the difference: Viruses, Worms, Trojans, and Bots? (2018, June 14). Retrieved December 30, 2018, from https://www.cisco.com/c/en/us/about/security-center/virus-differences.html#2

Wikipedia contributors. (2019, January 7). Samsung Galaxy S III. *In Wikipedia, The Free Encyclopedia.* Retrieved 04:00, February 9, 2019, from https://en.wikipedia.org/w/index.php?title=Samsung_Galaxy_S_III&oldid=877180916

Wikipedia contributors. (2019, January 17). Internet bot. *In Wikipedia, The Free Encyclopedia*. Retrieved 04:49, February 10, 2019, from https://en.wikipedia.org/w/index.php?title=Internet_bot&oldid=878849445

Wikipedia contributors. (2019, January 28). Computer worm. *In Wikipedia, The Free Encyclopedia.* Retrieved 02:57, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Computer_worm&oldid=880656399

Wikipedia contributors. (2019, January 31). Social engineering (security). *In Wikipedia, The Free Encyclopedia*. Retrieved 04:00, February 9, 2019, from https://en.wikipedia.org/w/index.php?title=Social_engineering_(security)&oldid=881038413

Wikipedia contributors. (2019, February 1). Adware. *In Wikipedia, The Free Encyclopedia.* Retrieved 01:28, February 7, 2019, from https://en.wikipedia.org/w/index.php?title=Adware&oldid=881243761

Wikipedia contributors. (2019, February 1). Phishing. *In Wikipedia, The Free Encyclopedia*. Retrieved 05:49, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Phishing&oldid=881233007

Wikipedia contributors. (2019, February 2). Ransomware. *In Wikipedia, The Free Encyclopedia.* Retrieved 12:25, February 7, 2019, from https://en.wikipedia.org/w/index.php?title=Ransomware&oldid=881383378

Wikipedia contributors. (2019, February 2). Stagefright (bug). *In Wikipedia, The Free Encyclopedia*. Retrieved 06:00, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Stagefright_(bug)&oldid=881497698

Wikipedia contributors. (2019, February 4). Spyware. *In Wikipedia, The Free Encyclopedia*. Retrieved 02:25, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Spyware&oldid=881734436

Wikipedia contributors. (2019, February 5). Android Oreo. *In Wikipedia, The Free Encyclopedia*. Retrieved 09:03, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Android_Oreo&oldid=881866871

Wikipedia contributors. (2019, February 6). Trojan horse (computing). *In Wikipedia, The Free Encyclopedia*. Retrieved 02:42, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Trojan_horse_(computing)&oldid=882070787

Wikipedia contributors. (2019, February 7). Android Pie. *In Wikipedia, The Free Encyclopedia.* Retrieved 09:05, February 8, 2019, from https://en.wikipedia.org/w/index.php?title=Android_Pie&oldid=882174758

Wikipedia contributors. (2019, February 7). Rootkit. *In Wikipedia, The Free Encyclopedia.* Retrieved 12:33, February 7, 2019, from https://en.wikipedia.org/w/index.php?title=Rootkit&oldid=882164080

Williams, A. (2018, August 06). *Android 9 Pie vs. Android 8 Oreo: What's new and what's changed?* Retrieved June 5, 2018, from https://www.techradar.com/sg/news/android-9-pie-vs-android-8-oreo-whats-new-changed-and-worth-nothing

Vania, J., Meniya, A., & Jethva, H. B. (2013). A review on botnet and detection technique. *International Journal of Computer Trends and Technology, 4*(1), 23-29.

Vegas, J. (n.d.). Gooligan. Retrieved February 2, 2019, from http://malware.wikia.com/wiki/Gooligan.

Verma, S. (2017, July 18). *Old Android exploit returns with new tricks to spy on you.* Retrieved May 28, 2018, from https://gadgets.ndtv.com/apps/news/trend-micro-ghostctrl-android-malware-worm-backdoor-1726355.

Vitre, P. (2018, September 12). *How to activate night mode on Android*. Retrieved January 8, 2019, from https://www.androidpit.com/night-mode-blue-filter-android

Yuhui, F., & Ning, X. (2014). The behavioral analysis of Android malware. Presented at the Next Generation Computer and Information Technology 2014 Conference. doi: http://dx.doi.org/10.14257/astl.2014.63.09