Summer 2018

Isolated Mobile Malware Observation

Augustine Paul
Georgia Southern University

Recommended Citation

1 Downer, K., and Bhattacharya, M.: 'BYOD security: A New Business Challenge', in Editor (Ed.)^(Eds.): 'Book BYOD security: A New Business Challenge' (IEEE, 2015, edn.), pp. 1128-1133 2 Brooks, C.: 'BYOD (Bring your own device)', Chartered Institute for IT–West London Branch.[Online]. Available at: http://www. bcs. org/upload/pdf/cbrooks-oct12. pdf.(Accessed: 22/03/13), 2012 3 Bello Garba, A., Armarego, J., and Murray, D.: 'BRING YOUR OWN DEVICE ORGANISATIONAL INFORMATION SECURITY AND PRIVACY ', ARPN Journal of Engineering and Applied Sciences, 2015, 10, (3), pp. 1279-1287 4 Romer, H.: 'Best practices for BYOD security', Computer Fraud & Security, 2014, 2014, (1), pp. 13-15 5 Lyon, C., and Osterman, M.: 'Security BYOD: Be Your Own Defense', in Editor (Ed.)^(Eds.): 'Book Security BYOD: Be Your Own Defense' (ACM, 2014, edn.), pp. 29-32 6 Wei, T.-E., Jeng, A.B., Lee, H.-M., Chen, C.-H., and Tien, C.-W.: 'Android privacy', in Editor (Ed.)^(Eds.): 'Book Android privacy' (IEEE, 2012, edn.), pp. 1830-1837 7 Jin, R., and Wang, B.: 'Malware Detection for Mobile Devices Using Software-Defined Networking', in Editor (Ed.)^(Eds.): 'Book Malware Detection for Mobile Devices Using Software-Defined Networking' (IEEE, 2013, edn.), pp. 81-88 8 Zaman, M., Siddiqui, T., Amin, M.R., and Hossain, M.S.: 'Malware Detection in Android by Network Traffic Analysis', in Editor (Ed.)^(Eds.): 'Book Malware Detection in Android by Network Traffic Analysis' (IEEE, 2015, edn.), pp. 1-5 9 Abdellatif, M., Talhi, C., Hamou-Lhadj, A., and Dagenais, M.: 'On the Use of Mobile GPU for Accelerating Malware Detection Using Trace Analysis', in Editor

ISOLATED MOBILE MALWARE OBSERVATION

by

AUGUSTINE PAUL

(Under the Direction of Christopher Kadlec)

ABSTRACT

The idea behind Bring Your Own Device (BYOD) it that personal mobile devices can be used in the workplace to enhance convenience and flexibility. This development encourages organizations to allow access of personal mobile devices to business information and systems for businesses operation. However, BYOD opens a firm to various security risks such as data contamination and the exposure of user interest to criminal activities. Mobile devices were not designed to handle intense data security and advanced security features are frequently turned off. Using personal mobile devices can also expose a system to various forms of security threats like malware. This research aims to analyze mobile network traffic from suspicious mobile applications and investigate data accessible to malicious applications on mobile devices. The research is further intended to observe the behavior of malware on mobile devices. A network with a wireless communication over a centralized access control point was built. The control access point serves as the centralized location for data monitoring, capturing and analyzing of transmitted data from all the devices connected to it. The research demonstrates a procedure for data capturing for analysis from a data collection point which does not require access to each application and allows for the study of potential infections from the outside of the mobile device.

INDEX WORDS: Mobile device, BYOD, Malware network traffic observation

ISOLATED MOBILE MALWARE OBSERVATION

by

AUGUSTINE PAUL

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in

Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

ISOLATED MOBILE MALWARE OBSERVATION

by

AUGUSTINE PAUL

Major Professor:	Christopher Kadlec
Committee:	Elizabeth Rasnick
	Lei Chen

Electronic Version Approved:
July 2018

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Dr. Christopher Kadlec for his valuable direction, encouragement and support in the progress of this research and thesis work. I would also express my most sincere gratitude and thanks to Dr. Elizabeth Rasnick and Dr. Lei Chen, members of the Thesis Committee, for their advice and support to the accomplishment of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

The trend in technological advancement is to enhance and integrate the robust functionality and high computing power of personal hand-held devices. The concept behind Bring Your Own Device (BYOD) is that personal mobile devices are to be used in the place of work to enhance convenience and flexibility. This innovation encourages organizations to allow access of personal mobile devices to enterprise data and systems for businesses operation.

A report from Downer, K., and Bhattacharya, M[1] indicated that about 70% of businesses are introducing BYOD which has brought about improved experience, enhanced productivity, efficiency, and reduced hardware expenses. In most organizations, the budget spent on technology is high. BYOD reduces costs and increases profits. BYOD shifts the responsibilities to individuals to purchase, and to maintain access of the devices. This reduces possible expenditure on hardware and maintenance from the organization's budget.[2] Additionally, BYOD offers the benefit of lower technical training costs for employers. A study by Cisco Networks (2012) indicated that companies reported an increase in employee productivity and efficiency due to implementation of the system of BYOD. For these reasons, many organizations started adjusting their IT system to implement BYOD [3].

The most vulnerable aspect of BYOD is in information security and privacy. Malware is one major threat to BYOD. Phishing and social engineering is another method devised by hackers to get information from employees using mobile devices and this is sometimes done forcefully. Hackers also use their skills to identify and analyze mobile devices and then launch an attack against such mobile devices.

Other BYOD security-risks include data contamination and the exposure of user habits to the activities of criminal syndicates. Mobile devices were not designed to handle rigorous data security and advanced security features are often turned off by default. The possibilities of data contamination by accidently or carelessly mixing personal files with business information could also be a threat to security, thus introducing malware into business file servers. New forms of malware targeting mobile devices are on the rise. Attacks range from mischievous pranks to advanced persistent threats that stealthily copy internal data over many months and

transmit it to remote control centers around the world. Also, attacks that would have been caught by defenses if devices are being used at work could make their way into the devices when users are actively off work.[4]

Data communication interception or spoofing is another threat to a wireless network used by employees' mobile devices. It occurs when users send information over the internet. Such information, when in the wrong hands, could be destroyed, modified or used in deceiving people (duping). Loss and theft of devices are some threats to IT security as mobile devices, thereby making the information they contain vulnerable. Malicious insiders could also be a problem to IT security. They could steal, modify or even destroy data from within the organization. User policy violation by employees could also make organizational data vulnerable, such as turning off firewalls and visiting dangerous websites. [3]

There have been various techniques which suggest how malware over a mobile device can be detected. These techniques require access to individual mobile devices, programs and installed applications. In addition, techniques involve a form of reverse engineering and a heuristic approach which could be intensive, resource consuming and difficult.

The objective of this study is to demonstrate how easy it is for a mobile device with no added form of protection, to be infected with a malicious program. It also seeks to demonstrate a way of isolating the mobile device on a network in order to capture and analyze what goes over it without exposing the network to the internet. This research aims to analyze mobile network traffic from suspicious mobile applications and investigate data accessible to malicious applications on mobile devices. The experiment is intended to further establish the network traffic behavior of malware on mobile devices. Subjects will utilize mobile devices for a limited amount of time, including installation of mobile applications (apps). It is believed by the researcher that electively installed apps, on behalf of the subjects, will try to transmit data out of the mobile devices. Any data transmitted out of the machines during the limited amount of time will be captured and analyzed for type of content. The devices will be cleansed of all personal data once the study is complete.

CHAPTER TWO

BACKGROUND

Mobile Malware Evolution

Mobile malware is believed to have evolved over time. As there is an increase in mobile phone usage and reliance over time, users become increasingly more susceptible to mobile malware threat and attack. The first mobile malware attack can be traced back to 2004. Since then, the number of attacks has increased rapidly, affecting many mobile phone users all over the world. Mobile malware can be grouped into four main categories across IOS and Android Operating Systems. The four main groups are: Trojans, Phishing attacks, hidden processes, and spyware. While they all operate differently, these malware attacks have a common goal: to extract personal data for monetary gain.

A Brief history to Mobile Malware

Early Discovery

The first malware was known as Cabir. Its operation was known to infect the most popular operating system of the time – Symbian. This malware caused the word Caribe to be displayed on the device's screen. Cabir spread itself to another device through enabled Bluetooth. This type of malware enables security data transfer of online activities. In its earliest introduction, it was used for tracking the activities of mobile phone users.

Monetary Gain

In 2010, there was an explosion of cybercriminals collaborating for pecuniary gain, exploiting vulnerabilities on smart phones. One popular example was Zitmo, also known as Zeus-in-the-mobile, which was a Trojan that could migrate between PC and mobile environments. This Trojan allowed online security processes, especially banking, to be explored. This led to massive loss of online transactions.

Android Market Domination

Cybercriminals took advantage of how popular the Android market had become. One specific Trojan developed to capitalize on this popularity was Droid Dream. It was uncovered in Google Play Store and where it had infected over 50 apps. Starting in 2011, Trojan attacks on Android phones became more pervasive, which led to important information being sent to third-parties and the unsolicited installation of unwanted apps. These

apps later infected the device. One year later, the first ever mobile ransomware emerged. Ransomware is a malware type that is designed to deny access to data until a sum of money is paid.

Continued Threat

Beginning in 2015, there has been an increase of mobile malware attacks. Experts determined that malware compromised more than 95 percent of all mobile devices between January and April 2016. An example is SMS Thief. This malware is sophisticated, hiding as an uninstaller malicious app. Mobile malware will only advance as malicious code becomes more difficult to handle with time. However, a few tips could help for protection against malware attacks. The mobile device user should keep apps up-to-date, avoid spam, keep browsing safe, and install antivirus software on their mobile devices.

Other types of Malware

Two other types of malware that cannot be categorize with other types of malware are drive-by malware and zero-day vulnerability. They are categorized separately because of the ways this malware infects devices. These types of malware expose mobile users to different forms of security risk. Drive-by malware is a threat that does not require any user interaction to download and execute malicious programs. They can be easily initiated either by clicking on a deceptive pop-up window or by viewing an email message. On mobile devices, drive-by malware can easily be used for data theft, installation of unsolicited malicious applications and to run commands that take control of the device remotely. Likewise, zero-day vulnerabilities are unknown vulnerabilities being exploited to adversely affect computer software. They are unknown because they are not publicly disclosed or reported. Through zero-day attacks, malware can infiltrate BYOD allowing unwanted access to user information. It can be used to orchestrate a distributed denial of service attacks. Drive-by malware and zero-day vulnerabilities are notable threats that could ravage businesses utilizing BYOD policies. [5]

Android Privacy

The rise in the use of smart phones has led to the emergence of mobile application services. Two of the most popular are the iPhone on App store and the Google Android market. These two application stores provide mobile users with a myriad of applications which they can download freely or for a token amount. These applications come with security threats, making security and application services important to smart phone users. Due to their popularity, iPhones and Androids are the most targeted by malware. Malware not only attacks

smart phones, but also extracts the user's private information. Malware often exploits the intrusion detection system (IDE) to scan viruses among it. A misuse-based method can detect known malware, but they have serious shortcomings with new malware.

Charlie Miller (2011) indicated differences between iOS (iPhone) and Android. The major differences identified are: code signing, app source, removed app, sandbox, and developers which limit malicious app activities on these respective smartphones. Applications can be downloaded from Android market without accessing a significant quantity of personal data. That is different from the App store. For app developers, the App store's method requires the developed app needed to use a private encryption key. Wetheral et al (2011) used web and smart phone apps. They unraveled how websites and smart phone apps access and obtain personal information when users do not have a notification of how personal information is exposed.[6]

In the following section, various attempts to isolate and study mobile malware are reviewed. This is done to understand the scope and extent to which previous work has contributed to this area of study. It also helps to identify a proper method to be implemented that will assist further contribution to knowledge.

CHAPTER THREE
LITERATURE REVIEW

The prevalence of mobile devices has attracted an increase in attacks and a growing number of mobile malware. Jin, Ruofan and Wang, Bing [7] reported that many devices were being infected through malicious applications available at different app stores. Being warned by security experts against mobile malware for years, their research focused on mobile malware. There are many reasons that have made mobile malware an increasingly greater threat. There has been a growing reliance and sole adoption of valuable information storage on mobile devices. This has sparked an increase of attackers. Connections to insecure networks with mobile devices are done with ease. Unpatched and out-of-date software/firmware of mobile devices make it easy to exploit vulnerabilities.

According to Jin, Ruofan and Wang, Bing [7] software defined networking (SDN) enhances efficient and flexible manners to achieve network security. In view of this, analyses based on mobile malware behavior was made, a mobile malware detection algorithm was proposed, designed and implementation of malware was executed utilizing SDN as a method. The outcome identifies suspicious network events through real-time traffic analysis. The execution of detecting algorithm insider open flow controller enables security rules to be imposed in real time.[7] Although Jin, Ruofan and Wang, Bing [7] claims to have achieved an extensive result, the process may not be persistent and pervasive since it requires further study on characteristics of mobile malware and how to explore detection within the context of SDN. We consider the method inappropriate to our study because we are not interested in reengineering an established procedure for malware detection on mobile devices. However, we want a process that will quickly access and analyze what is transmitted over a network.

Zaman, Siddiqui, Amin, & Hossain [8] demonstrated a detection method based on network traffic analysis. They discussed the different types of malware detection techniques and their effectiveness with specific types of malware. They thoroughly demonstrated a behavioral detection method for detecting mobile malware that can communicate with backlisted domains and pass sensitive personal/financial information. The method they used is effective against malware that communicates with known malicious remote servers. However, the detection method is limited to known behavior, pattern and signatures of malware on mobile

devices. The method implemented delves into how codes were written for each malware. The study considers this method inappropriate because it falls short of a quick approach to response to malware.

A typical security measure is the use of anti-malware, which is effective by matching data streams generated on mobile devices against known malware signatures. The drawback is that it limits and hinders scalability of mobile anti-malware systems because it reduces memory and computing power of mobile devices. Abdellatif, M., Talhi, C., Hamou-Lhadj, A., and Dagenais, M. [9] used an alternative to process detection solutions on an external server. However, this exposes the detection techniques and system to connectivity problems. The use of mobile Graphics Processing Unit (GPU) paired with anti-malware on mobile device, exploits the computational power of mobile GPU and increases memory optimization as the solution increases. [9] The method they applied required high consumption of hardware resources to process malware signature to achieve an effective detection and trace analysis. A significant shortcoming to their method would be a failure in any of the hardware required renders the entire process ineffective.

Mobile apps frequently transmit sensitive information through the network with different intentions. A few transmissions are required to satisfy the application's functionalities. Although such transmissions with malicious receivers may prompt security leakage and tend to go unnoticed. Fu, H., Zheng, Z., Bose, S., Bishop, M., and Mohapatra, P [10] recommended LeakSemantic as a system that can consequently find anomalous and delicate transmissions from mobile applications. It is comprised of a hybrid program analysis and a machine learning segment. The program analysis consolidates static and dynamic analysis to correctly distinguish sensitive transmissions. When compared with other newest analysis, LeakSemantic accomplishes better result with less false positives and can also gather runtime information compared to .[10]

The security feature integrated into the Android operating system helps to reduce and mitigate attacks by limiting applications using permissions and sandbox. Therefore, an attacker uses a deceptive means to introduce a malicious app to a mobile device. App installation requires that privileges are granted for requested permission. This remains unchanged until they are revoked when the app is removed. A proposed approach for malware detection is to leverage on both frequency of permission usage in malware and scarcity of them in normal apps. This approach can apply to filter suspicious apps for further analysis. By implementing the approach, evaluation shows a higher detection rates and is more effective.[11] The process

may not necessarily be effective in a situation where a quick incident response is needed to detect malware behaviors and activities.

Network administrators need to be aware of applications running in their system. This is foundational for both security and network administration. Recent years have seen an exponential development in the quantity of mobile applications which has complicated this undertaking. Conventional strategies for traffic arrangement are not any more adequate as the larger part of mobile phone application traffic is conveyed over HTTP/HTTPS. Being aware of the new applications that surface regularly is extremely tasking and tedious. A novel procedure was implemented that automatically created network profiles for recognizing Android applications in HTTP traffic. A network profile is comprised of fingerprints, i.e., one of a kind attributes of network conduct that can be utilized to recognize an application. To profile an Android application, Dai, S., Tongaonkar, A., Wang, X., Nucci, A., and Song, D [12] ran the application naturally in an emulator and gathered the system that follows.

Gorla, Tavecchia, Gross, & Zeller's [13] approach effectively identifies applications whose behavior would be unexpected given their description. This identification was made known through the clustering apps by description topics and identifying outliers by app usage within each cluster. They identified several examples of false and misleading advertising; and as a side effect, obtained a novel effective detector for yet unknown malware. Gorla, Tavecchia, Gross, & Zeller[13] discovered that mining apps and their descriptions open several new opportunities for automated checking of natural-language requirements. They gained several insights into the Android app ecosystem that call for action.

Zhu, Xiong, Ge, & Chen[14] developed a mobile App recommender system with security and privacy awareness. Without depending on any predefined risk functions, they designed a scalable and automatic approach for estimating the security risks of mobile Apps. The mobile App recommender is a framework developed to detect and evaluate the security and privacy risk of mobile Apps automatically. A unique perspective of this approach is the creative use of external knowledge as prior scores and the regularization techniques in an App permission bipartite graph. To consider both App's popularity and users' security preferences for recommendations, Zhu, Xiong, Ge, & Chen[14] introduced a flexible apps recommendation method based on the modern portfolio theory. An App hash tree was developed to efficiently look up Apps in

recommendation. Their experiments on a large-scale real-world data set clearly validated the effectiveness and efficiency of the proposed recommendation framework.[14]

Lin, Amini et al [15] discovered that both users' expectation and the purpose of why sensitive resources are used have a major impact on users' subjective feelings and their trust decisions. They also found that informing users properly on the purpose of resource access can ease users' privacy concerns to some extent. Based on their findings, common user misconceptions were highlighted about privacy and Android permission. Their findings interface is much easier to understand, and it provides users with more pertinent information for users to make better trust decisions.[15]

There is also a combined analysis of static and dynamic type for malware detection on Android. The process enables static analysis to check the application on the phone before setup, after which is passed for dynamic analysis if any problem is detected. Dynamic analysis uses specific tools to analyze files, determine any suspicious actions and then inform the user of the result.[16] This approach consumes hardware resources and requires reverse engineering for malware detection. This is considered less effective for a quick incident response for mobile malware detection in that it investigates what is in each mobile device and how permission and functions are written.

Bayer, Kirda, &Kruegel [17] recommended a novel method for making the dynamic analysis of malicious programs more efficient. It considerably reduced time expanded for analyzing a set of malicious programs. They detected that a program was a polymorphic variation of an already analyzed binary by executing it for a short period of time. They checked whether the behavior seen was identical to an already analyzed binary executed within the time frame allotted. In the future, Bayer, Kirda, &Kruegel [17] propose a plan to actively use this technique in its dynamic analysis system because it helps the study to analyze more of today's malicious programs. The process requires an extensive heuristic approach and a reverse engineering method which is difficult when a quick response is needed. Since applications and programs are usually written differently and the study's intended interest is on what goes across network, this method is considered inappropriate for this study.

Gilbert, Chun, Cox, & Jung[18] presented a vision for implementing automated security validation of mobile apps at app markets. They proposed AppInspection, a security validation service that analyzed smartphone apps submitted to app markets and generated reports that aided in identifying security and privacy risks. It sketched the high-level design of AppInspector and discussed several challenges and ideas for approaching them. Gilbert, Chun, Cox, & Jung[18] opined that large-scale automated validation of apps at a central distribution point was an important step towards enabling more secure mobile computing, and therefore advised the research community to take advantage of the opportunity. In view of this, the study considers the approach limited and non-pervasive in that inspecting individual mobile applications is a difficult task which does not offer a fast response to malware detection and analysis.

Mobile malware analysis is evolving, and the study considered the methods implemented in various literature. Although these methods were effective within the scope of all the literatures that have been reviewed, those methods are considered inappropriate for a quick approach to analyze malware on a mobile device. Some of the methods implemented in the literature reviewed, requires reengineering an established procedure for malware detection on mobile devices. Other methods require the use of hardware. This study is interested in demonstrating how to isolate possible infected mobile device and capture the traffic generated at data collection point.

CHAPTER FOUR

METHODOLOGY

The purpose of our research was to present a fluid analytical approach that is effective in analyzing traffic of devices on a network without looking into what apps are on the devices itself. Additionally, this approach does not allow infections to spread while the device is being analyzed. This approach to malware analysis is quick and effective. A network is set up that isolated potentially infected devices, then performed an analysis of the traffic transmitted out of the device. This method did not involve any form of heuristic or reverse engineering approach and took no interest looking into what was in the device. This made the analysis more fluid and effective. Additionally, this uncovered behavioral activities of malwares on mobile devices. Security details on malware behavior, analysis and detection was inevitable. The study described the experimental process executed to analyze and investigate mobile malware through various mobile applications. The study's focus on mobile platform was Android OS. The reason for our choice was because Android was one of the largest mobile platforms with countless applications available to its users. Another reason was Android operated an open platform which allowed malicious app developers to exploit their vulnerable users.

The research procedure involved various activities that exposed and infected the sample mobile devices with malware and analyzed acts of installed programs over an established network with packet capturing tools. To achieve our intended result, an experimental network topology was created (see Figure 1). This network topology was based on wireless communication over a centralized access control point. The network topology in Figure 1 consisted of mobile devices used for a week and a control access point that appeared like a wireless access point to all connected mobile devices. Appendix A has the full description of the mobile device. This control access point was our centralized location for data monitoring, capturing and analysis. The network setup was conceived and implemented to have control access over all devices and their transmitted data. Also, the setup was necessary to contain and properly capture transmitted packets of the mobile devices connected to the control access point.

The study requested voluntary participation of individuals; participants were notified by an individual e-mail that they were selected for the study, consent was obtained, and a device was delivered to them. The details information for the informed consent is shown in Appendix B. Participants were asked to utilize mobile

devices that were provided for a period of one week. Five graduate students of the Department of Information Technology were chosen as participants. The recruitment was from the pool of graduate student of the department of information Technology. This is a purposely built pool because these individuals are familiar with mobile devices and mobile software. They are not randomly selected students and are not representative of the general public. Because of the procedure that was implemented, a large amount of data was required to carry out this study. Hence, the participants are not being analyzed, but the behavior of the software that they install. The selection was based on the response received from the email that was sent out to them. The participants' use of these devices involved exploring every functionality, not limited to downloading and installation of applications alone, but also surfing the internet. After a week, all devices were retrieved from all participating personnel at which point their involvement ended. The devices were then connected to our access control point for further monitoring and analysis. The study looked at the behavior of the devices after the week, and not the behavior of the participants. Participants were not associated with the data from the devices once they were returned and the devices were only differentiated by the specific device, and not who might have used it. The control point was a private secure network that did not transmit to the Internet. The control point was a computer with wireless access enabled, that appeared as a wireless access point to the mobile devices and served as a central location to capture intended traffic. This one control point was the data collection device.
To monitor traffic, Wireshark, a network sniffing software was installed on the control point and used for data collection. The data collected from the devices was primarily protocols established for transmission between the device and the intended, but not accessible, Internet destination. There was no direct analysis of what was done on the device and the study only analyzed how the device tried to send out data once they had been collected. The devices used for the study were cleaned by restoring them to factory defaults. This ensured an absolute erasing of data on the device.

The study focused on patterns and behavior of malicious programs installed on mobile devices, how it transmitted data over the network, and what access and privilege was granted to mobile apps. Also, a key interest was on information mobile apps transmitted over network connections. The study setup a local wireless connection where all understudied mobile devices were connected. Hence, with packet capturing tools, the study analyzed the behavior and pattern of installed programs on these devices for suspected malicious acts.

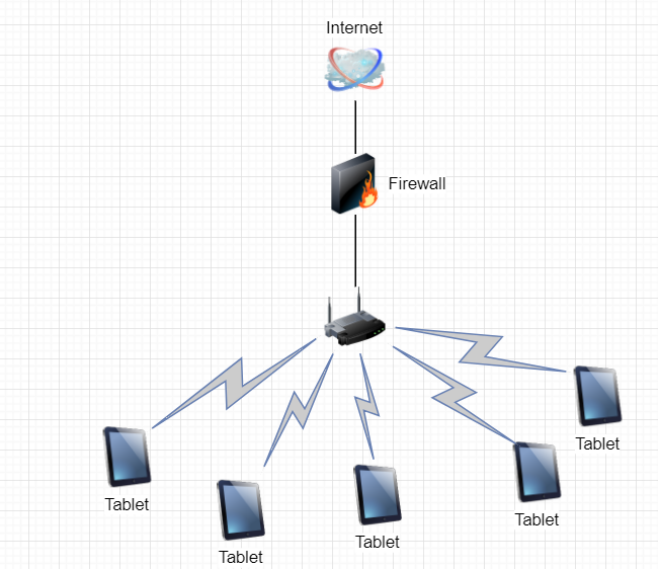

Figure 1 Standard Network Architecture.

The researcher's network Figure 1 was secure and private with a control access point at the core of it. The access control point was a computer with fully functioning wireless features enabled. It served as the data collection point. All traffic from the mobile devices was transmitted through the access control point and captured with a packet sniffing software. Analysis was then performed based on the traffic generated to further study the behavior of malware on these mobile devices. The access control point was built on a Dell Inspiron computer with functioning wireless card that supported WI-FI access point infrastructural mode. A Linux operating system was installed on the computer with other packages that aided the configuration of the machine and its wireless card. An access point package was installed and configured (see Appendix C). This package made the computer wireless card act as WI-FI access point for the local network. This setup allowed a total control of the network and removed the stress of acquiring a separate wireless router.

To further establish a proper implementation of a WI-FI network, the researcher configured a computer that acted as our access control point configure with iptables, DNS and DHCP (see Appendix C, D, and E). This was done to achieve a fully functional wireless access point that communicated with devices over the network. The iptables, a firewall installed on most Linux distributions, consisted of chains of rules which allowed traffic to be forwarded through identified ports associated with defined protocols. The DHCP configuration was primarily for automatic IP assignment to all connected mobile devices.

The researcher decided that DNS calls may influence the study in that a mobile device may try to query the DNS, a response to such query allows tracking of domains visited over the network. The study allowed a DNS call by configuring a caching DNS, a temporary database maintained on the control access point. The caching DNS contained records of all the recent visits and attempted visits to the internet domains. The purpose of the caching DNS was to resolve queries from the mobile devices connected to the control access point, to handle the tracking of such queries, and to respond to such queries.

The study demonstrated a way of isolating a network connected with possible infected mobile devices while still allowing the traffic to be generated. This method gave the freedom to perform packet capturing and analysis. To know what kind of traffic was being generated from each device, IP addresses were reserved to match the MAC addresses of the mobile devices used for this research. This also helped to track calls, requests, queries and answers to each mobile device. The caching DNS passed DNS requests and answered DNS calls. The configuration of the caching DNS aided our study by making name resolution possible, which further allowed devices to proceed with an attempt to communicate over the network.

Establishing A Baseline Traffic Analysis

Data for analysis was captured from the data collection point that is the controller access point. This was the data collected after the device had been restored to its factory default and free of any kind of installations. Figure 2 is the traffic representation capture with Wireshark which is constituted of protocol such as TCP, DNS, ARP, and EAPOL.

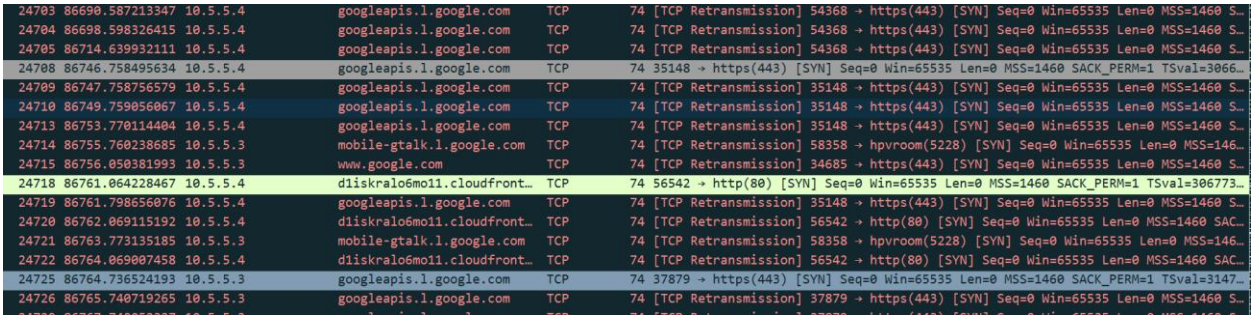

Figure 2 Baseline Traffic on Wireshark

Almost all the TCP and DNS traffic were communication the devices tried to make to the pre-installed Google apps, API and Google public DNS server. There were four layers in the Transmission Control Protocol/Internet Protocol TCP/IP protocol architecture. The layers were Application, Transport,

Internet, and Network Interface. The network interface layer controlled how TCP/IP packets were placed on the network medium and received packets off the network medium. The Internet layer was responsible for addressing, packaging, and routing functions. The main protocols that operated on the Internet layer were Internet Protocol (IP) responsible for addressing, routing, fragmentation, and reassembly of packets; Address Resolution Protocol (ARP) responsible for resolution of IP address to hardware address; Internet Control Messaging protocol (ICMP) responsible for diagnostic functions, error reporting due to connectivity issue of IP packet, and Internet Group Messaging Protocol (IGMP) manages membership of multicast group. The Transport layer established session and datagram communication services for the Application layer. Its main protocols were Transmission Control Protocol (TCP) which made possible peer-to-peer, connection-oriented reliable communications service and User Datagram Protocol (UDP) which also provided a peer-to-peer or peer-to-many unreliable communications. TCP provided apps a way to send and receive an ordered and error-checked stream of information packets over the network. UDP was used by apps to transmit a faster stream of information by doing away with error-checking. The Application layer was used for data exchange and allowed various applications to have access to services provided by other layers. Two examples of protocols which operated in this layer were Domain Name Systems (DNS) and Hypertext terminal protocol (HTTP). The TCP/IP provided basic protocol installed in every network operation system which aided interconnection of other protocols and other applications relied on the service provided by these core protocols. The core protocols were ARP, ICMP, IGMP, TCP, UDP, and IP.

Other noticeable protocols generated on our network were Dynamic Host Configuration Protocol (DHCP), Network Time Protocol (NTP), Multicast Domain Name System (MDNS), Extensible Authentication Protocol Over LAN (EAPOL), and BJNP. DHCP was used in Internet Protocol networks to dynamically assign an IP address to devices on a network from a configuration server. The DHCP was configured on the server in such way that allowed for manual pre-configuration of MAC address to IP address mappings for each client. EAPOL an IEEE 802.1X authentication protocol was developed for a generic network sign-on access network resources. It established an authentication conversation using a simple encapsulation over any LAN. NTP protocol is used to synchronize the clocks of computers over the internet. MDNS resolved host names to IP addresses within small networks that excluded a local name server. It

operates as a semantics unicast DNS with a zero-configuration service, utilizing basically similar programming interfaces, and packet formats.

The need for baseline traffic is to define the underlying network activity of the devices, helping to identify changes in our network, allowing for quick analysis. Table 1 shows a protocol distribution on the mobile network traffic for a time frame of 24 hours generated from Wireshark and the total occurrence of protocols made by all the devices on the network. The study noted that whenever the device slept every form of communication stopped for at least for 30 minutes. Though the connection to the control point was not lost, each device's wireless radio was rendered temporarily inactive; at this point activities across the network seem to be halt.

Protocol	
TCP	198,886,157
ARP	65,323,038
DNS	21,302,262
EAPOL	19,183,037
MDNS	616,288
NTP	293,216
DHCP	186,880
BJNP	102,318
ICMP	20,870

Table 1 Baseline Protocol Distribution

CHAPTER FIVE

RESULTS

The study was carried out using Wireshark, a control access point and five mobile devices. This procedure demonstrated a way of isolating a mobile device on a network to capture and analyze mobile network traffic from suspicious mobile applications and investigated data accessible to malicious applications on mobile devices.

The study established a baseline traffic that defined the original state of our network. It was a snapshot of the mobile network traffic for a time frame of 24 hours generated from Wireshark. The baseline consists of traffic that reveals the type of activities inside the mobile devices. This is important to understand changes to network activities which can be of help for analysis. Table 2 is the post manipulated traffic generated from Wireshark. This table shows that there are some additional protocols introduced to the network (highlighted in grey), this shows the kind of activities in each of the mobile device on the network.

Protocol	
TCP	27,905,209,303
ARP	2,633,730,689
DNS	1,495,864,364
EAPOL	216,998,253
QUIC	112,202,246
NTP	89,198,790
MDNS	2,913,377
ICMP	2,863,033
DHCP	2,640,345
BJNP	290,632
BASICXID	77,697

Table 2 Post Manipulated Protocol Distribution

The difference between Table 1 and Table 2 shows a mark increase of traffic generated as seen in Table 3. Comparing the baseline with the post manipulated reading indicates a huge percentage raise of traffic utilization for all the devices on the network. The informational data in Table 3 is the difference between post manipulated traffic and baseline. The percentage increase shows the differential of generated traffic between baseline and post manipulated traffic.

Protocol	Baseline Reading	Post Manipulated Reading	Informational Data	Percentage Increase
TCP	198,886,157	27,905,209,303	27,706,323,146	1393074%
ARP	65,323,038	2,633,730,689	2,568,407,651	393186%
DNS	21,302,262	1,495,864,364	1,474,562,102	692209%
EAPOL	19,183,037	216,998,253	197,815,216	103120%
QUIC		112,202,246	112,202,246	
NTP	293,216	89,198,790	88,905,574	3032085%
MDNS	616,288	2,913,377	2,297,089	37273%
ICMP	20,870	2,863,033	2,842,163	1361841%
DHCP	186,880	2,640,345	2,453,465	131286%
BJNP	102,318	290,632	188,314	18405%
BASICXID		77,697	77,697	

Table 3 Informational Data

These values show the amount of traffic that was generated with addition of three mobile applications to all the mobile devices. Applications such as Instagram, Innovator, Twitter, and some other games applications were installed on some of the mobile device. On each mobile device, three applications were installed and explored for a duration of one week. Table 4 reveal various traffic activities that were generated of a single mobile device on the network. This shows tones of negotiation of TCP, DNS and other protocols generated.

Protocol	Baseline Reading	Post Manipulated Reading	Informational Data	Percentage Increase
TCP	39,777,231.40	5,581,041,861	5,541,264,629	1393074%
ARP	13,064,607.60	526,746,138	513,681,530	393186%
DNS	4,260,452.40	299,172,873	294,912,420	692209%
EAPOL	3,836,607.40	43,399,651	39,563,043	103120%
QUIC	-	22,440,449	22,440,449	
NTP	58,643.20	17,839,758	17,781,115	3032085%
MDNS	123,257.60	582,675	459,418	37273%
ICMP	4,174.00	572,607	568,433	1361841%
DHCP	37,376.00	528,069	490,693	131286%
BJNP	20,463.60	58,126	37,663	18405%
BASICXID	-	15,539	15,539	

Table 4 Informational Data for a Single Mobile Device

Table 5 shows traffic snapshots of the mobile devices with various intended endpoints. From the table, the transmission (Tx) packet and byte generated by mobile devices on the network reveals the type of activities from the mobile devices. Packet transfer from mobile device to the resolved addressed in Table 5 failed because

the network was design not to send out or receive traffic. This demonstrate that infected device can be isolated on a network to capture traffic generated for analysis.

Address	Tx Packets	Tx Bytes	Country
2.android.pool.ntp.org	1	90	Germany
star.c10r.facebook.com	6	444	Ireland
instagram.c10r.facebook.com	8508	629592	Ireland
d1iskralo6mo11.cloudfront.net	30	2220	U.S. A
d1iskralo6mo11.cloudfront.net	30	2220	U.S. A
device-metrics-us-2.amazon.com	596	44104	U.S. A
device-metrics-us-2.amazon.com	576	42624	U.S. A
d1iskralo6mo11.cloudfront.net	756	55944	U.S. A
d1iskralo6mo11.cloudfront.net	756	55944	U.S. A
api.amazon.com	5	370	U.S. A
api.amazon.com	10	740	U.S. A
device-metrics-us-2.amazon.com	400	29600	U.S. A
2.android.pool.ntp.org	3	270	U.S. A
api.amazon.com	5	370	U.S. A

Table 5 Intended Endpoints communications

Also, the result as seen in Table 6 shows how each mobile device is trying to transmit packet across the network to various addresses. Though the setup was established to block packet transmission outside of the network, each mobile device constantly tries to transfer packets. From the table, about forty-one thousand packets were generated by all the mobile devices on the network. This shows the type of activities that each mobile device tries to initialize. The resolved addresses, packet size and protocols indicated certain types of conservations from each device, this information is important to our analysis because it identified the type of traffic generated and allow for tracking various IP addresses involved in the traffic exchange.

10.5.5.6	42014	android.l.google.com	https	5	370	5	370
10.5.5.6	36048	android.l.google.com	https	4	296	4	296
10.5.5.6	41496	android.l.google.com	https	4	296	4	296
10.5.5.5	39684	instagram.c10r.facebook.com	https	3	222	3	222
10.5.5.5	51854	android.l.google.com	http	4	296	4	296
10.5.5.5	39685	instagram.c10r.facebook.com	https	5	370	5	370
10.5.5.5	39686	instagram.c10r.facebook.com	https	5	370	5	370
10.5.5.5	39687	instagram.c10r.facebook.com	https	5	370	5	370

Table 6 Conversations

The result was important for the study because it shows an aim to capture traffic for analysis of mobile network from suspicious mobile applications and investigate data accessible to malicious applications on mobile devices. The study shows that, there is a large traffic generated by all the devices on the network. This indicated what was going inside each of these devices. Also, it demonstrates that possible spread of malicious software can be contained by isolating the network of suspicious infected devices.

CHAPTER SIX

CONCLUSION

This study presents how to quickly acquire a snapshot of mobile network traffic for analysis of intended malicious activities or behavior. It also demonstrates an effective means to capture traffic for analysis without looking into the devices itself. The study shows how to capture traffic for analysis that does not require going into each application and allow to study potential infections from the outside of the mobile device. This setup is crucial because it ensures that viruses and malware does not spread across the network. This setup does not require a whole lot of hardware utilization because it allow for a quick approach for malware analysis that does not require expensive hardware. The study does not require heuristic or reverse engineering approach. The study does not require a reengineering an established procedure for malware detection on mobile devices. The study builds a network with a wireless communication over a centralized access control point. The control access point was implemented to serve as the centralized location for data monitoring, capturing and analysis for all devices and their transmitted data. The study confirms that this approach is a fluid process by which a quick analysis can be done on a network to uncover suspicious activities on a network. It also established a quick process to access and capture for analysis what is transmitted over a network as well as identified traffic increase on the network. The study has shown captured traffic from data collecting point connected to the mobile devices with various intended endpoints, which is of important because it identified the kind of traffic generated and allow for tracking various IP addresses with their locations involved in the traffic exchange.

REFERENCES

1 Downer, K., and Bhattacharya, M.: 'BYOD security: A New Business Challenge', in Editor (Ed.)^(Eds.): 'Book BYOD security: A New Business Challenge' (IEEE, 2015, edn.), pp. 1128-1133

2 Brooks, C.: 'BYOD (Bring your own device)', Chartered Institute for IT–West London Branch.[Online]. Available at: http://www. bcs. org/upload/pdf/cbrooks-oct12. pdf.(Accessed: 22/03/13), 2012

3 Bello Garba, A., Armarego, J., and Murray, D.: 'BRING YOUR OWN DEVICE ORGANISATIONAL INFORMATION SECURITY AND PRIVACY ', ARPN Journal of Engineering and Applied Sciences, 2015, 10, (3), pp. 1279-1287

4 Romer, H.: 'Best practices for BYOD security', Computer Fraud & Security, 2014, 2014, (1), pp. 13-15

5 Lyon, C., and Osterman, M.: 'Security BYOD: Be Your Own Defense', in Editor (Ed.)^(Eds.): 'Book Security BYOD: Be Your Own Defense' (ACM, 2014, edn.), pp. 29-32

6 Wei, T.-E., Jeng, A.B., Lee, H.-M., Chen, C.-H., and Tien, C.-W.: 'Android privacy', in Editor (Ed.)^(Eds.): 'Book Android privacy' (IEEE, 2012, edn.), pp. 1830-1837

7 Jin, R., and Wang, B.: 'Malware Detection for Mobile Devices Using Software-Defined Networking', in Editor (Ed.)^(Eds.): 'Book Malware Detection for Mobile Devices Using Software-Defined Networking' (IEEE, 2013, edn.), pp. 81-88

8 Zaman, M., Siddiqui, T., Amin, M.R., and Hossain, M.S.: 'Malware Detection in Android by Network Traffic Analysis', in Editor (Ed.)^(Eds.): 'Book Malware Detection in Android by Network Traffic Analysis' (IEEE, 2015, edn.), pp. 1-5

9 Abdellatif, M., Talhi, C., Hamou-Lhadj, A., and Dagenais, M.: 'On the Use of Mobile GPU for Accelerating Malware Detection Using Trace Analysis', in Editor (Ed.)^(Eds.): 'Book On the Use of Mobile GPU for Accelerating Malware Detection Using Trace Analysis' (IEEE, 2015, edn.), pp. 42-46

10 Fu, H., Zheng, Z., Bose, S., Bishop, M., and Mohapatra, P.: 'LeakSemantic: Identifying Abnormal Sensitive Network Transmissions in Mobile Applications', in Editor (Ed.)^(Eds.): 'Book LeakSemantic: Identifying Abnormal Sensitive Network Transmissions in Mobile Applications' (IEEE, 2017, edn.), pp. 1-9

11 Deypir, M.: 'A new Approach for Effective Malware Detection in Android-based Devices', in Editor (Ed.)^(Eds.): 'Book A new Approach for Effective Malware Detection in Android-based Devices' (IEEE, 2016, edn.), pp. 112-116

12 Dai, S., Tongaonkar, A., Wang, X., Nucci, A., and Song, D.: 'Networkprofiler: Towards automatic fingerprinting of Android apps', in Editor (Ed.)^(Eds.): 'Book Networkprofiler: Towards automatic fingerprinting of Android apps' (IEEE, 2013, edn.), pp. 809-817

13 Gorla, A., Tavecchia, I., Gross, F., and Zeller, A.: 'Checking app behavior against app descriptions', in Editor (Ed.)^(Eds.): 'Book Checking app behavior against app descriptions' (ACM, 2014, edn.), pp. 1025-1035

14 Zhu, H., Xiong, H., Ge, Y., and Chen, E.: 'Mobile App Recommendations with Security and Privacy Awareness', in Editor (Ed.)^(Eds.): 'Book Mobile App Recommendations with Security and Privacy Awareness' (ACM, 2014, edn.), pp. 951-960

15 Lin, J., Amini, S., Hong, J.I., Sadeh, N., Lindqvist, J., and Zhang, J.: 'Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing', in Editor (Ed.)^(Eds.): 'Book Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing' (ACM, 2012, edn.), pp. 501-510

16 Su, M.-Y., Fung, K.-T., Huang, Y.-H., Kang, M.-Z., and Chung, Y.-H.: 'Detection of Android Malware: Combined with Static Analysis and Dynamic Analysis', in Editor (Ed.)^(Eds.): 'Book Detection of Android Malware: Combined with Static Analysis and Dynamic Analysis' (IEEE, 2016, edn.), pp. 1013-1018

17 Bayer, U., Kirda, E., and Kruegel, C.: 'Improving the Efficiency of Dynamic Malware Analysis', in Editor (Ed.)^(Eds.): 'Book Improving the Efficiency of Dynamic Malware Analysis' (ACM, 2010, edn.), pp. 1871-1878

18 Gilbert, P., Chun, B.-G., Cox, L.P., and Jung, J.: 'Vision: automated security validation of mobile apps at app markets', in Editor (Ed.)^(Eds.): 'Book Vision: automated security validation of mobile apps at app markets' (ACM, 2011, edn.), pp. 21-26

APPENDICES

APPENDIX A
MOBILE DEVICE DESCRIPTION.

The mobile device in this research is Google Nexus 7 tablet. It runs an Android 5.1.1 and powered by 1.5Ghz quad-core processor. It RAM size is 2GB and has an internal storage capacity of 23GB. The battery is non-removable. Google Nexus 7 has connectivity options that includes Wi-Fi and Bluetooth. This device was cleansed first by rebooting into bootloader with Android debug bridge(adb) command. While in the bootloader a fastboot oem unlocked was used to reset the device. The fastboot unlocked option allow for the reinstallation of the operating system. This procedure cleans up the device and allows for a fresh installation of Android 5.1.1.

APPENDIX B
INFORMED CONSENT

My name is Augustine Paul, a graduate student of the department of Information Technology. I am using this study for my thesis requirement toward achieving the specified degree program.

The Purpose of this study is to analyze mobile malware activities that might have associated with the applications you have installed on the mobile device.

The procedures: In this study, participants will be given a mobile device to use for one week. During that week, they can use the device however they like to download applications and access the internet. At the end of the week, the device will be returned and analyzed for malicious applications. Prior to accepting the device, we will ask for to consent to the study. We will be primarily looking at graduate students in technical fields to maximize the possibility that they will download applications or apps. We will be looking at the behavior of the devices after the week and will not need the participants to do anything more. The devices are to be connected to a control point that will monitor the behavior of the device and not the participants. The control point will be a private secure device that will not transmit to the Internet. The control point is a computer with wireless access enabled, that appears as a wireless access point to the mobile devices. It will serve as a central location to capture intended traffic. This one control point will be the data collection device. Our main research interest and focus is on the behavior of the mobile devices that will be given to participants. Mainly, the research will investigate patterns and behavior of the mobile device and installed applications after they are returned. Devices for this study will be issued to participants at the department of Information Technology. Participants are at liberty to download applications of their choice. These devices will be in the custody of the participants for a week. The participants can use it at home or wherever he/she is permitted to access the internet. After which the device will be retrieved then connected to the control point for monitoring. Participants will be ensured that the devices will not be probed. After the device has been retrieved, monitoring activities begin. On the control point, Wireshark, a network sniffing software is used for data collection. The data collected from the devices is primarily protocols established for transmission between the device and the intended, but not accessible, Internet destination. There will be no direct analysis of what was done on the device and we will only analyze how the device tries to send out data once they have been collected. The devices used for the study will be cleaned by restoring them to its factory defaults. This ensure an absolute erasing of data on the device.

Researcher may be able to gain access to personal information about the subjects that would have been left on the devices. Any of such personal information will be treated with alter most confidentiality. Also, the mobile devices will be cleansed of all changes made by participants after the study. Only the behavior of the devices will be captured by the control point and all data captured will be cleansed of personal data.

The benefit of this study is to broaden on the effect and behavioral impact of malware on mobile devices both to individual users and organization, so that security measure against malware attack will be properly implemented.

The duration is one week from the date approve of consent is agreed.

Your information will be treated with optimal confidentiality. Devices will be held on a private network and will not be able to send information out of that network. It is the attempts to send information that will be captured and analyzed. Once the study is over the device will be cleaned, wiped and restored to factory default. The captured data will be cleansed of any personal data.

Please do direct all corresponding regarding this research to the researcher or the researcher's faculty advisor, contact information provided below. For questions concerning your rights as a research participant, contact Georgia Southern University Office of Research Services and Sponsored Programs at 912-478-5465.

There is no compensation in any form involve in this study. Participation is voluntary.

Your decision to participate in this research work is voluntary. You may at any time withdraw interest or may decide to discontinue your involvement in this study but will have to return any issued device in your custody. Participant information will be treated with optimal confidentiality. Devices will be held on a private network and will not be able to send information out of that network. It is the attempts to send information that will be captured and analyzed. Once the study is over the device will be cleaned, wiped and restored to factory default. The captured data will be cleansed of any personal data. The identity of participant for this study is not attached to any of the device. Each device has a physical (MAC) address used to reserve an IP address. This is the only ID that will be associated to the device and no record of participant will be attached to the device.

There is no penalty for deciding not to participate in the study; you may decide not to participate in this study at any time you want and may withdraw without penalty or retribution.

You must be 18 years of age or older to consent to participate in this research study. If you consent to participate in this research study and to the terms above, please sign your name and indicate the date below.

This project has been reviewed and approved by the GSU Institutional Review Board under tracking number H18233.

Title of Project: Mobile Malware

Principal Investigator: (Augustine Paul, Department of Information Technology, ap07667@georgiasouthern.edu)

Faculty Advisor: (Dr. Christopher Kadlec, Department of Information Technology, ckadlec@georgiasouthern.edu)

_____ _____

Participant Signature Date

I, the undersigned, verify that the above informed consent procedure has been followed.

_____ _____

Investigator Signature Date

APPENDIX C
HOSTAPD CONFIGURATION ON LINUX

Check whether your Wi-Fi card support AP

lspci -k | grep-A 3 -i "network"

check interface details of the wireless driver for compatibility

modinfo ath9k | grep'depend'

Install Hostapd

sudo apt-get update &&sudoapt-get installhostapd

or download hostapd and compile it

configuring hostapd

open hostapd.conf file in /etc/hostapd-2.6/hostapd directory and add the configuration below to the file

```
interface=wlp9s0
driver=nl80211
ssid=ResearchAP
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=KeePGuessinG
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

APPENDIX D
SETTING UP THE DHCP SERVER

Install dhcp
Sudo apt-get update &&sudo apt-get install isc-dhcp-server

Edit dhcpd.conf

```
ddns-update-style none;
ignore client-updates;
authoritative;
option local-wpad code 252 = text;

subnet
10.5.5.0 netmask 255.255.255.248 {
# --- default gateway
option routers
10.5.5.1;
# --- Netmask
option subnet-mask
255.255.255.248;
# --- Broadcast Address
option broadcast-address
10.5.5.7;# --- Domain name servers, tells the clients which DNS servers to use.
option domain-name-servers
10.5.5.1;
option time-offset
0;
range 10.5.5.2 10.5.5.6;
default-lease-time1209600;
max-lease-time1814400;
}
```

APPENDIX E

CREATE A BASH SCRIPT TO INITIAL THE DHCP, ENABLE NAT AND LAUNCH WLAN INTERFACE

```
#!/bin/bash
#Initial wifi interface configuration
ifconfig$1 up 10.5.5.1 netmask 255.255.255.248
sleep2
###########Start DHCP, comment out / add relevant section##########

#Doesn't try to run dhcpd when already running
if[ "$(ps -e | grep dhcpd)"== ""]; then
dhcpd $1 &
fi
###########
#Enable NAT
iptables --flush
iptables --table nat --flush
iptables --delete-chain
iptables --table nat --delete-chain
iptables --table nat --append POSTROUTING --out-interface $2 -j MASQUERADE
iptables --append FORWARD --in-interface $1 -j ACCEPT

sysctl -w net.ipv4.ip_forward=1
#start hostapd
hostapd /etc/hostapd-2.6/hostapd/hostapd.conf 1>/dev/null
killalldhcpd
```

launch the hostapd in the background using the command

sudohostapd -B /etc/hostapd-2.6/hostapd/hostapd.conf

grant the bash script an executable writes

sudochmod +x researchsoftAP

Run the script

Sudo ./researchsoftAP