IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ANALYSIS AND SECURITY TESTING OF ANDROID SYSTEM BY MALWARE DETECTION IN NETWORKS

Deepika Vaghela
Master of Computer Engineering, Parul Instituteof Engineeringand Technology, India

ABSTRACT

The same principles that organizations use to monitor network traffic go into their networks must be applied to the network traffic originating from mobile devices. This means that the techniques and tools, which would normally be used to collect and analyze network activity, can also be used to detect anomalous network traffic or network intrusions related to smartphones. This report will therefore outline an architecture model, which can be used to analyze the network communications originating from Android devices and to detect any unusual traffic. As part of the exercise, a set of several tests involving real malware will be executed to gauge the effectiveness of said architecture. In addition to that, the aim of the exercise is to improve the detection mechanisms of the engine by creating new signatures to detect specific threats.

KEYWORDS:A ndroid Application, SQL server, Network traffic monitoring,malware

INTRODUCTION

While mobile devices have grown in popularity among consumers, they have also become very attractive to malicious application developers. Malicious applications, or malware, are applications which negatively affect the function of a device, steal sensitive data, or attain unconsented control of said device. The wide usage of mobile devices generates an ever increasing amount of sensitive personal data, all of which is vulnerable to loss or theft. Between 2012 and 2013, the number of unique malware samples that were detected has increased by 300%; these samples were found in nearly 4 million application installation packages. As the world's most popular mobile operating system, Google's Android OS is the principal target of an ever increasing mobile malware threat. To counter this emerging menace, many malware detection techniques have been proposed.

There are two broad methods of malware detection - static analysis and dynamic analysis. Static analysis, as the name suggests, consists of methods which do not require an application to be executed (i.e. code review). In contrast, dynamic analysis is a behavioral study of an application at runtime. Each technique has its own advantages and disadvantages. While static analysis is the most energy efficient and simplest to implement, malware developers have discovered techniques to make their applications more difficult to detect using this method. On the other hand, dynamic analysis has the potential to provide more thorough detection due to its runtime analysis. It is, however, very resource intensive and is difficult to implement and automate.

Hear proposed dynamic "Network traffic" analysis as a supplemental approach to detect malicious applications which evade static analysis. Through small-scale experiments, we found that communications between suspicious IP addresses can be captured and analyzed using a packet. To address this issue it should be explore a dynamic analysis technique that monitors network traffic using a packet sniffer which provides insights for dynamic malware detection.

Android Malware

As the mobile devices technology evolve to a high-end state and able to support complex OS, it has become the malware next target. With the high proliferation of Android OS in the market, the malware targeting Android or refer as Android malware are also rising.

Since the first discovery of Android malware known as Fake player in 2010; malware for Android has grown to an alarming rate. The malicious activities executed by the malware can varies, for instance Fake player is a trojan that hide behind a legitimate movie player application and can sent SMS messages automatically to premium rate number without the user knowledge (Castillo, 2013). Geimini, Pjapss and Hippo SMS are other example of Android malware that capture or send SMS without the user knowledge. Some other Android malware are more sinister, it can exploit the Android vulnerabilities and gain an administrator or root access, Droiddream, DroidKungfu and BaseBridge are an example of Android malware that come with privilege escalation binary for exploiting Android vulnerabilities. Zhou and Jiang (2012) have investigated 1260 Android malwares and identify each of the malware malicious behaviour.

The study shows Android malware can be categorized based on their malicious payload, which are privilege escalation, remote control by contacting command and control server, Financial charges by automatically send SMS to premium number and information theft by stealing information such as device information, phone number, contact list and GPS location information.

Stealing device and user credential information

Most of the Android malware are collecting information regarding International Mobile Equipment Indentity (IMEI) number, International Mobile Subscriber Identity (IMSI) number, GPS Location, Phone number, SDK version and Installed package. Android malware also captured user activities such as SMS send or received and call out or in duration.

Communicate with Command and Control (C and C) server

Android malware have the intention to communicate to a C and C server similar to Botnet. Communication can be made through HTTP or SMS. This connection are made for sending captured information from the device and also for updating the malware package or receiving any other malicious information for instance SMS premium number or any other malicious URL.

Sending premium rate SMS and spam: Android malware has the ability to automatically send SMS without the user knowledge. This can be used as a spam or gathering illegal income by charging user when the SMS is send to the premium number.

Search engine optimization

Android malware can have an automatic script to visit ad, for increasing the number of visitor to a website or for the purpose of generating a charge per click without having actual interest in the target of the ad's link. Even though, it seem like it does not cause any harm, this activity can increase mobile data consumption especially to the user who are subscribing for the mobile broadband it will cause them extra charges.

Updating and download package

Android malware can used the communication made to the C and C server to update the existing package into a more malicious intention or even download a new package and installed it automatically in the user devices.

Draining resources

Android malware application can executed a malicious payload that can utilize all the disk storage or memory (RAM) quotas and hogging the CPU.

The increase of Android malware on the market has been substantially disturbing, every new malware on the market come with new features, ability and new intention. The malicious intentions above might only be valid with the current Android malware but as the advancement of mobile technologies is growing in a rapid pace, it is possible that the consequences of Android malware will be more devastating to the user. Thus a more effective solution must be developed especially in combating a zero-day Android malware.

Survey on Android malware detection methods in networks

In 2012 , Detecting Application Update Attack on Mobile Devices through Network Features describes that cross-feature analysis approach tries to solve the classification problem Thus, an ensemble of learners for each one of the features represents the model through which each new vector of features is tested for "normality". For the online analysis, each one of the instance features is predicted by the corresponding classification\regression model using the values of all other features. The more different the predictions are from the true values of the corresponding features, the more likely that the observed instance comes from a different distribution than the training set (i.e., represents an anomaly event).

In 2012 , Towards a framework for Network-based Malware detection System . In this work we present a Network-based Malware detection System to provide situational awareness of mobile devices connected through a VPN network. Its goal is to combine network traffic analysis with signature-based IDS to provide both the mobile user and the network administrators with alarms of anomalous behaviour and means for visualization analytics of mobile devices.

In 2013 , Research report , Using NIDS/NIPS to detect malware on Android mobile devices , The approach is to utilize Snort open source (NIDS/NIPS), to inspect the network trace of the Android device. Snort is the most utilized (NIDS/NIPS) and utilize a rule-driven
language that combine the techniques of signatures and anomaly based inspection methods, the signature is used to detect patterns of specific known exploits and vulnerabilities. An approach of inspecting the Android device traffic on a Virtual Private Network (VPN) Server running Snort is presented by Parrizas (2013) [2], that's the one will be follow The goal is to implement a design that would store the VPN Server alarms generated by Snort in a database and design an Android application that will give the VPN Clients the information about the alarms generated by their mobile devices.

In 2013 , Malware Detection for Mobile Devices Using Software-Defined Networking , Whenever a packet comes to an OpenFlow switch, it will be compared against the flow tables in the switch. If a matching flow entry is found, the actions associated with the flow entry will be executed. Otherwise, the packet will be forwarded to the controller, which then decides how to deal with the packet and installs flow entries to the switch if needed. So it is not other than a behavior based analysis software –defined networking.

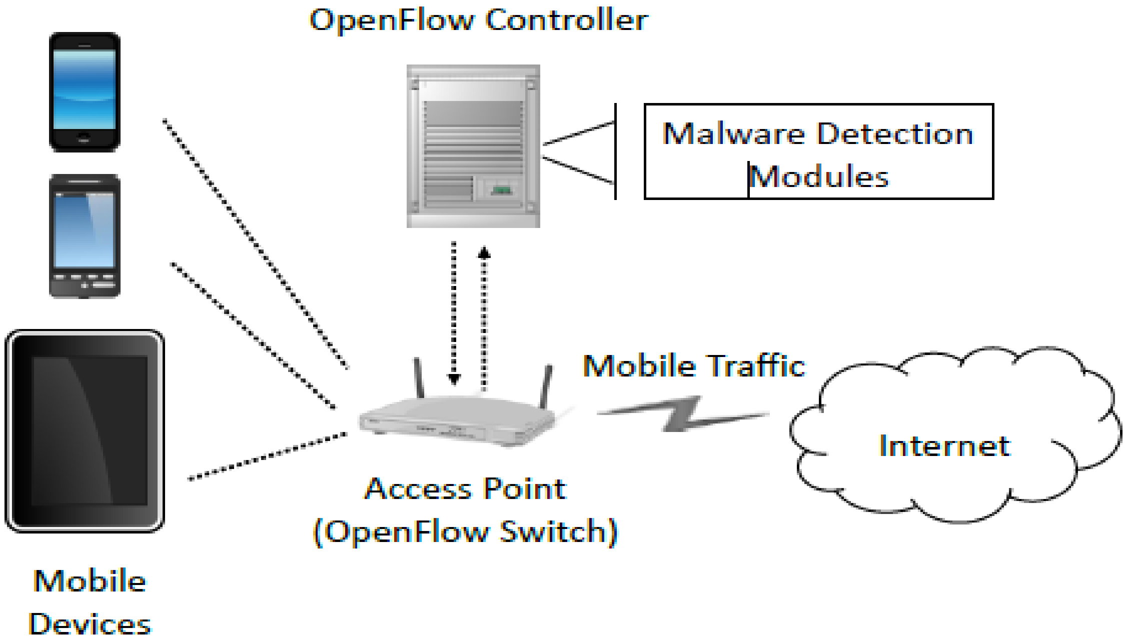

Figure 1 . Architecture of malware detection system

In 2014, Research of Android Malware Detection Based on Network Traffic Monitoring , It causes is malicious sample concentration of some software link server have failed, and some malicious software was unable to complete the network communication. The reason of false alarm is too little training data set, in order to reduce the rate of false positives; we need more data extraction for a long time to complete the training.

In 2014, Research of Android Malware Detection Based on Network Traffic Monitoring , It causes is malicious sample concentration of some software link server have failed, and some malicious software was unable to complete the network communication. The reason of false alarm is too little training data set, in order to reduce the rate of false positives; we need more data extraction for a long time to complete the training.

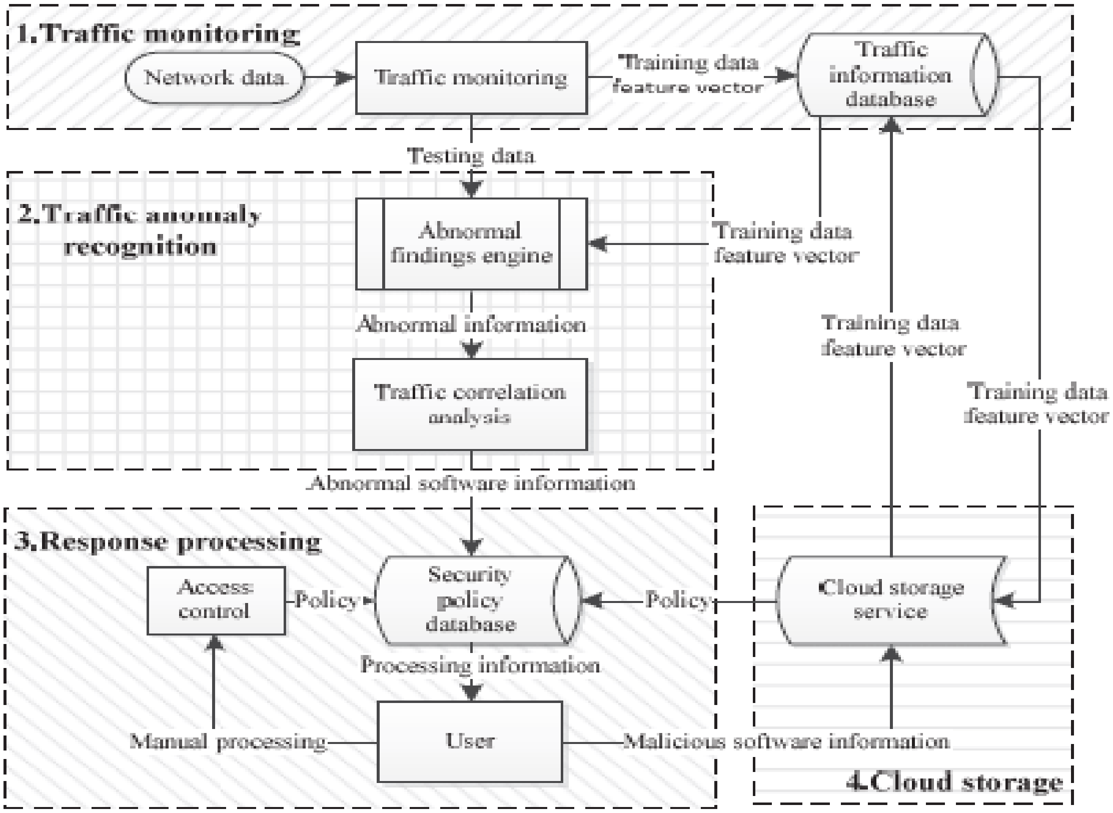
1.Traffic monitoring
Network data
Traffic monitoring
Training data feature vector
Traffic information database
Testing data
2.Traffic anomaly recognition
Abnormal findings engine
Training data feature vector
Abnormal information
Training data feature vector
Traffic correlation analysis
Training data feature vector
Abnormal software information
3.Response processing
Access control
Policy
Security policy database
Policy
Cloud storage service
Processing information
Manual processing
User
Malicious software information
4.Cloud storage

Figure 2 . Network traffic monitoring system framework

These methods can be found by malicious software to network traffic anomaly of Android system, in order to further analysis result in abnormal software, also need the network traffic anomaly information and software information, and locate the application that produced abnormal through the correlation analysis.
The main drawback is it is host based, time consuming & depends on the system's configuration.

LIMITATION OF EXISTING METHODS

An organization should ensure that it has a clearly defined privacy policy that spells out the rights of individuals using its network, grants permission to sniff traffic for security and troubleshooting issues, and states the organization's policy requirements for obtaining, analyzing and retaining network traffic dumps which will be limited to our project also, but the Existing methods still not so helpful to find zero day malware which specific antimalware software signatures are not yet available.

CONCLUSION & FUTURE WORK

Throughout this project, we have been able to deploy a network architecture which permits to closely monitor the network flows originating from the smartphone. This model permits an Intrusion Analyst to monitor the network connections in real time and review the flows in order to perform forensic analysis. With this setup, we have been able to analyze several pieces of malware, test the effectiveness of our IDS and learn how the alerts are generated. This analysis has permitted to improve the existing Snort signatures for Android.

REFERENCE

1. Mohd Zaki Mas`ud, Shahrin Sahib, , Mohd Faizal Abdollah, Siti Rahayu Selamat and Robiah Yusof, 2014. Android Malware Detection System Classification. Research Journal of Information Technology, 6: 325-341. Portokalidis et al., "Paranoid Android:
2. Versatile Protection for Smartphones," Proc. Ann. Computer Security Applications Conf. (ACSAC 10) ACM, 2010, pp. 347-356.
3. Android Architecture Overview. http://developer.android.com/images/ system-architecture.jpg.
4. Android Behind The Scenes: Revealing Hidden Malware With Andromeda By Robert Joseph Marsan.
5. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-Based Malware.
6. Detection System for Android," Proc. ACM Workshop Security and Privacy in Mobile Devices (SPMD 11), ACM, 2011, pp.
7. Dupaul, N. "Common Mobile Malware Types: Cybersecurity 101", Oct 2013. http://www.veracode.com/blog/2013/10/common-mobile-malware typescybersecurity-101/
8. Mohd Zaki Mas`ud, Shahrin Sahib, , Mohd Faizal Abdollah, Siti Rahayu Selamat and Robiah Yusof, 2014. Android Malware Detection System Classification. Research Journal of Information Technology, 6: 325-341.
9. Android Developers - Content Providers. http://developer.android.com/guide/topics/ providers/content-providers.html.
10. Android Developers - Intents. http://developer.android.com/reference/android/ content/Intent.html.
11. Mobile malware evolution: 3 infection attempts per user in 2013, February 2014. Accessed 17 July 2014.
12. Manilyzer: Automated Android Malware Detection through Manifest Analysis http://nlab.engr.uconn.edu/papers/manilyzer-camera-ready.pdf
13. Practical Malware Analysis ,Kris Kendall, kris.kendall@mandiant.com