WallDroid : Firewall for the Android OS

Abhijeet Awade, Amir Talwar, Bhushan Khopade and Vishal Nande

B.E Computer Engineering, Navsahyadri Education Society's Group of Institutions , Pune.

Abstract -

Security is becoming an increasingly important feature of today's mobile environment where users download unknown apps and connect their smartphones to unknown networks while roaming. Android has become a very popular operating systems for smartphones and tablets but at the same time threats associated to this platform, like malware or exploits, are also growing. As it is becoming more and more popular to connect mobile phone and other hand held devices to the internet, the big question is; "How to protect those devices from the perils of the internet?". This project investigates issues with the implementation of a Firewall system for protecting mobile devices. To enable an ordinary mobile phone user to setup a Firewall configuration to protect his mobile phone it is important to have a system that is easy to understand and warns the user of possible mistakes.

This project proposes and valuates an enhanced security model and architecture, WallDroid, enabling virtualized application specifc Firewalls. The WallDroid solution can be considered as an Android Firewall Application but with some extra functionality. Key components used by the solution include VPN technologies like the Point to Point Tunneling Protocol (PPTP) and the android Cloud to Device Messaging Framework (C2DM). Our project is based on the cloud keeping track of millions of applications and their reputation (good, bad, or unknown) and comparing traffic flows of applications with a list of known malicious IP servers.

Keywords : Android OS; Security; Mobility.

1. Introduction

Nowadays, anyone can implement Android applications without having strong programming skills. So the cost of developing Android applications is very low. Most companies and developers do not have the proper security skills, to create secure Android applications. Therefore, the developers sometimes do not consider all security issues or more often, they are simply not skilled enough to be aware of all vulnerabilities [1].

It is the developer who specifies which permissions the application requires. Then, when the user installs the application, the user is presented with a list of the developer's requested permissions. The user must grant all permissions, otherwise, the application will not install. The allowed permissions cannot be changed at run time. Once the application is installed, it may obtain or give other applications sensitive data. Applications can also obtain sensitive data, by interacting with the user. The sensitive data might be shared between a normal application and a malware application. Then the malware application may transmit that sensitive data via the Internet directly. Again, if the malware application does not have direct access to the Internet, if may access the Internet indirectly, via a normal application. Malware applications and even normal applications may communicate sensitive information via Internet servers or via SMS/MMS without notifying the user [3].

To prevent leaking personal data and react fast, we suggest a framework, which aims to provide secure connections by normal and even malicious applications.

2. Existing system

Applications in Android devices may be installed through either the Google Play Store (formerly called Android Market) or through a variety of third party application stores. Some of the third party stores are Opera Mobile app store, GetJar, SlideME, etc. The fact that Android permits application installation from third party vendors means that Google has no control over the quality or safety of the applications provided in these stores. Several cases were encountered where legitimate apps from the Google Play Store were modified to inject malicious code and the modified apps were sold in these third party stores. It is difficult to determine whether the application is genuine or not. In these cases, the reliability of the application depends upon the security measures implemented by the application store. This makes it essential to provide a reliable means to verify the authenticity of the applications[2]. In case of existing system the Android OS uses multiple 'Anti-Virus' for avoid Malware.

Google's Play Store has a security enforcement known as the "Bouncer" which verifies the applications being uploaded for any suspicious behavior. It works on a blacklisting based approach and instances have occurred where-in the malicious application survived several hours or even days before it was detected and taken off. However, this app still does not protect the users from installing malicious apps from sources other than the Play Store. 4.2 version (Jelly Bean) of Android has shipped with a security feature to counter this problem. A new feature allows the user to verify the third party application being installed on the phone[2].

3. Problem statement

"Design an Application which provides enhanced security to the Android Phones from Malwares and Data Leaking."

4. Proposed system

In this system we are providing Android based Firewall, give permission or denied permission to Application for Wi-Fi & Data usage. This will be advantages if any Un-known process try to execute on Mobile without user permission then application catch & Block it. This will secured Mobile phone from Execute any Harmfull process on phone[1].This application is a combination of multiple security utilities. We have seen that there is no single app at Google Market/ Google Play which not only informs the user about the spyware attack on his privacy but also provide feature of mobile Anti-Malware. So, this app is not an antivirus; but a Mobile Firewall. WallDroid is a bunch of utilities which will alert the user whenever some malicious activity takes place on his mobile or tablet[3]. Our app will inform the user whenever such malicious activity happens on his mobile. This will help the user to know whether his mobile is being used by other app or not[2]. For e.g. if you have buyed a new mobile its battery drop must not be frequent, but if it happens then your mobile have been probably infected by Malware.

5. Background

A. The Android Operating System

Google's Android OS is built on the ARM platform with a modified Linux kernel of version 2.6.x (versions older than 4.0) or 3.x (version 4.0+) and was released in 2008. Lower level system utilities are written in C while most user "applications" are written in Java, although it is possible to write applications using native code in a language such as C++ . Google's custom Dalvik virtual machine is a replacement for the standard Java virtual machine used on other desktop and server platforms; the engine is optimized for limited resources typically available on mobile devices. Therefore, if a developer writes an Android application in Java, it will be compiled into dex bytecode (in a file called classes.dex), not standard Java bytecode that would run on Windows or UNIX platforms. A detailed presentation of Dalvik can be found in Dalvik author Dan Borstein's presentation.

The most interesting feature about Android is that the kernel places each application in a sandbox when it executes. This isolates the application from all the other applications and other parts of the operating system. This involves the use of standard UNIX process separation techniques which allow the application to access its stored data and memory without being able to interfere with the other applications hardware, memory and data usage. Each application is assigned a unique UID (user ID) and GID (group ID). User can install their choice of applications from the Google Play Store or they can directly install them in the memory card. While installing the applications, the user is presented with certain permissions requested by the application like access to the Internet, access to GPS coordinates, accessing contacts, etc. The user can either choose to accept all permissions requested by the application or choose to not install the application[1].

5.System Requirement

A) Hardware Interfaces:

For successful working of the discovery module the following elements are the important requirements:

i. Mobile (Android OS)

B) Software Interfaces:

Operating System	: Windows, UNIX
Front End	: Java
Data Bases	: My Sql
IDE	: Eclipse(ADT)

Java (JDK 6)

Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

JSP

Java Server Pages (JSP) technology enables Web developers and designers to rapidly develop and easily maintain, information-rich, dynamic Web pages that leverage existing business systems. As part of the Java technology family, JSP technology enables rapid development of Web-based applications that are platform independent. JSP technology separates the user interface from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content.

MySql Server 5.1

MySQL is a popular choice of database for use in web applications. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools,[citation needed] or download MySQL front-ends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structures, and work with data records.

C) Communication Interface

i. Touch-pad
ii. Internet
iii. Monitor

6. System Architecture:

The architecture of our application based security model, which is called WallDroid.

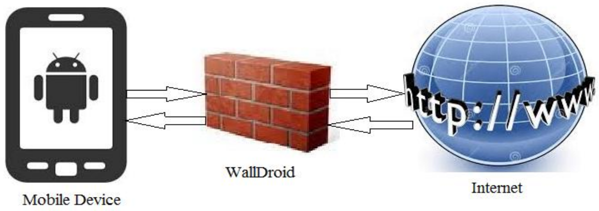

Figure 6.1 Solution Architecture

The architecture of our application based security model, which is called WallDroid. The aim of WallDroid is to detect malicious activity at a very early stage and then to quickly prevent any future malicious activity. The WallDroid architecture consists of three main components:

1) Mobile device.
2) WallDroid app.
3) Internet.

The WallDroid app can be considered as an Android Firewall Application but with some extra functionality. Before presenting more details concerning our solution, we will prevent various anti-malware strategies, which we believe are inferior, to our solution. Some anti-malware solutions require the user to decide what to do, for applications which are not clearly safe and not clearly malware. However, the user is often not in the best position to make a decision. We therefore propose that the user choose a security policy. There could be a large number of different security policies, which the user could subscribe to. However, we will greatly simplify the security policy discussion and just mention a few examples.

One anti-malware strategy is to grant permission to all Unknown applications. Another anti-malware strategy is to deny permission to all Unknown applications. The problem with these strategies is that these are far too general. The component is, also as mentioned above, the WallDroid Application Storing a table of applications, including their status and other statistics. Since vendors can use the same certificate for multiple applications and updates, we must first find a way to create our own unique application ID. Application is identified It is harmfull or not by its certificate which provide by Play-store.

we have three classifications of Android applications.

1) "The Good" - We have applications which are known to be good. For these applications, we grant permission for these good applications to have direct Internet access.

2) "The Bad" - We have applications which are known to be malicious (bad). For these applications, we deny permission for these bad applications to have direct Internet access. We also attempt to have these uninstalled.

3) "The Unknown" - The very interesting case is for applications which are not known to be good and not known to be bad.

These unknown applications are the focus of our solution. The component of our proposed solution is the WallDroid application. Part of the WallDroid application is a database service. It is this database service which contains the list of all applications and their reputations (good,bad and unknown) which WallDroid has ever encountered, on any user's Android. When WallDroid is installed, it sends the list of installed application hash values to the cloud (based on a subset of the applications' extracted files). It is then the database that returns the reputation of each installed application. If WallDroid detects any application-ID, which is not in the database service, it tags that application as Unknown.

WallDroid allows the Known-Good App to access Internet and connect its server directly without any limitation. It blocks the Known-Bad Apps by restricting permissions of that malicious app. When WallDroid determines an Unknown App, it automatically turns on the Android Data service and establishes an Internet connection. An according connection is established WallDroid System are also able to observe the Unknown app's data traffic to determine whether it is malicious app or the app is sending any personal data as a clear text. Once if WallDroid System figures out that the Unknown app is malicious or it does not care about network security, Application blocks the data traffic and informs the WallDroid Application.

6.2 Workflow of System

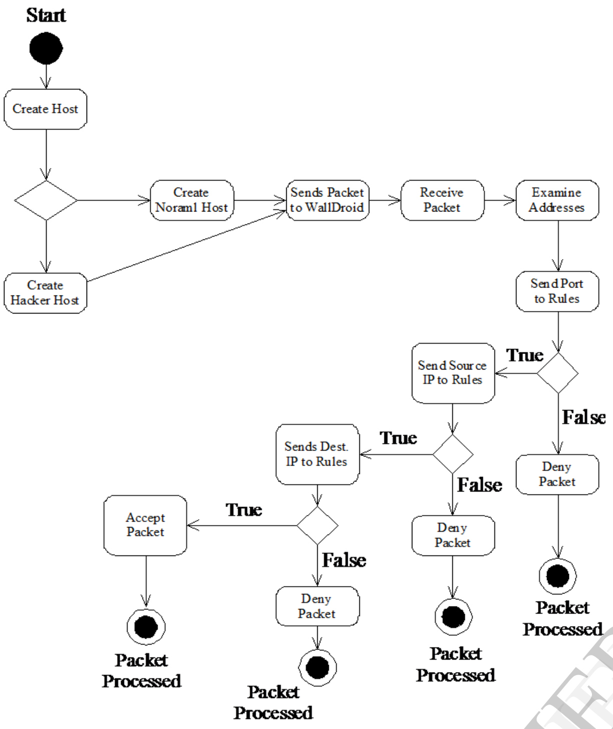

Figure 6.2 Workflow of System

The user must grant all permissions, otherwise, the application will not install. The allowed permissions cannot be changed at run time. Once the application is installed, it may obtain or give other applications sensitive data. Applications can also obtain sensitive data, by interacting with the user. The sensitive data might be shared between a normal application and a malware application. Then the malware application may transmit that sensitive data via the Internet directly.

The workflow or proposed system is as follows, If any packet want to enter into mobile or exit from mobile, it always goes through the WallDroid. After receiving packet WallDroid starts to examine that packet. It checks port number, source IP address and destinations IP address. If all this are found true then and then only that packet is accepted otherwise that packet will be discarded.

The apps which are not from Google play store, it means from third party will not contains any trust certificate, such applications also blocked or denied. The apps which has suspicious behavior like accessing contact information, message data or any system file again and again and without taking user's permissions, such apps declared as a Malicious apps and it also blocked for avoiding data leaking.

7.Advantages

- Provides security to Android Phones.
- Restrict unauthorized Processes.
- Anti-Malware.
- Filtering Packets.
- Prevent leaking of data.
- Allow or Block Internet Access to particular application

8. Conclusion

Existing System are too general. We therefore provide an ApplicationID-based solution for enhancing security which is a very fast mechanism in terms of actions being taken before any data is leaked. WallDroid works as Firewall as well as Ani-malware application.

9.References

[1] "WallDroid: Cloud Assisted Virtualized Application Specific Firewalls for the Android OS", Caner Kilinc, Todd Booth, and Karl Andersson, *2012 IEEE 11th International Conference.*

[2] "Malware Analysis for Android Operating", Kriti Sharma, Trushank Dand, Tae Oh and William Stackpole , *JUNE 4-5, 2013, ALBANY, NY.*

[3] "AndroidLeaks : automatically Detecting Potential Privacy Leaks In Android Applications on a Large Scale", Clint Gibler1, Jonathan Crussell1*;2, Jeremy Erickson1;2, and Hao Chen1.*

[4] "Reputation Based Security Model for Android Applications", Welderufael Berhane Tesfay, Todd Booth, and Karl

Andersson, *2012 IEEE 11th International Conference.*

[5] “Performance Modeling and Analysis of Network Firewalls”, Khaled Salah, Member, IEEE, Khalid Elbadawi, Member, IEEE, and Raouf Boutaba, Fellow, IEEE. *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 9, NO. 1, MARCH 2012*