Hierarchical Expert System for Security Evaluation and its Implementation on an Android Smartphone

Shivang Bokolia
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
sb8392@rit.edu

***Abstract*—Technology has advanced to a significant level in the past decade. A technology that has seen the most improvement would definitely be the "smartphones". With this advancement came two major developers of smartphone OS namely Android and iOS. Due to the tremendous development and growth in mobile phone software and hardware technologies, security has become a very big challenge to almost all concerned individuals – Scientists, manufacturers, designers, developers, etc. This paper discusses the security evaluation and methods for Android OS. It will discuss the current security features in Android along with a metric system that is used for evaluating the security and risk in an Android OS. Along with these it also discusses the implementation of an Expert System in an Android Device. For bonus, the paper elaborates the Android OS architecture, along with the different forms of threats and risks in Android Security. It also explains in detail the working of an Expert System along with an extended explanation of the different components for security evaluation in an information system.**

***Index Terms*—Android; Android OS; Android Architecture; Expert Systems; Metrics; Hierarchical Expert System; Security**

I. Introduction

Today, almost every individual walking on this planet has a mobile device and in the majority of cases, it is a smartphone. The smartphone market has grown dramatically in the past few years because of the ease they provide and eventually has become the platform of choice for users and different businesses. These small devices provide a broad range of services that include communication, internet browsing, entertainment, social media, financial processes, and much more which opened a new world of applications ad new possibilities, a world that provides comfort to the user regardless of their location. In addition, many of the traditional PC operations can now be performed on these smartphones as well. There are several players in the market that are providing these large ranges of features that include Samsung, Microsoft, Apple, Google, etc. but the main ones that develop the OS for all these devices are Google (Android OS) and Apple (iOS). Android OS and Apple iOS have about 99% of the global market share for operating systems. As of the recent data provided, there are about 3.8 billion smartphone users in the world out of which 72.19% are using Android OS and 27% are using Apple iOS which the rest are using other OS like Samsung, KaiOS, etc. [1]

Due to the significant growth of these devices, the amount of data generated from these has also increased significantly. Since data has recently become the new gold, there are several security threats that have generated that are ready to exploit the smart environment and use this data illegally. These security threats include information theft, identity theft, personal data intrusion, illegal access of the applications, distribution of malware, and financial fraud. [2]

Since the release of Android OS in 2009, many security solutions for smartphones have been proposed in the industry and the main focus has been on malware detection, host-based intrusion detection, access control, static analysis of applications, and encryption and isolation. The major security components of the Android OS security framework involve permission mechanisms, application sandboxing, and application signing. [3]

This paper presents three different sections related to Android smartphone and security and risk evaluation of the same:

i. Android OS Architecture and current security and risk evaluations employed.
ii. Metrics involved in security and risk analysis of the Android smartphone along with a design of the

hierarchical metrics system.

iii. Implementation of an Expert System in Android smartphone followed by the conclusion which describes the content for Project 2 – the application and methods involved in building the application, and Project 3 – the building of the Machine Learning model for approximation of the Expert System.

II. Android OS Architecture

Since the release of Android OS in 2009, there have been 17 different versions and out of the current population of Android OS users, 12.44% are Android 11.0 users, 38.63% are Android 10.0 users, 18.02% are Android 9.0 "Pie" users, 9.17% are Android 8.1 "Oreo" users, and the rest use OS older than the mentioned ones. Android OS also has various modifications made by hardware platform manufacturers as well and these may range from pre-installed applications to modifications in the OS kernel. This variety is called Android OS fragmentation. In addition, because of the Android OS open-source nature, various enthusiasts produced numerous unofficial versions. [3]

Android is an open-source, Linux-based software stack created for a wide array of devices and form factors. Android Architecture contains different number of components to support any android device needs: Applications, Applications Framework, Android Runtime, Platform Libraries, and Linux Kernel. Figure 1 shows the main components of the Android platform:

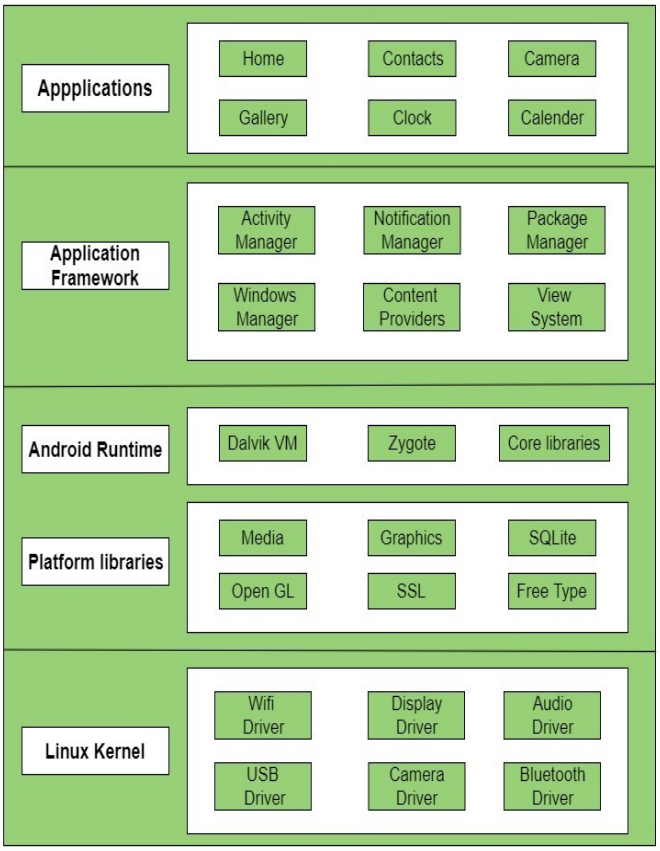

Fig. 1. Android OS Architecture

1) Applications:
The applications layer consists of the pre-installed applications like phone, contacts, camera, gallery, etc., and third-party applications downloaded from the play store are installed on this layer only.

2) Application Framework:
Applications Framework provides several important classes which are used to create an Android Application. It includes different types of services like activity manager, notification manager, view system, package manager, etc. It provides generic access for hardware access and also provides a user interface with the application resources.

3) Application Runtime:
Android Runtime Environment is one of the most important part of the Android system. It contains components like core libraries and the Dalvik virtual machine (DVM). It provides the base for the application framework and powers the applications through the core libraries.
Dalvik Virtual Machine (DVM) is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently.

4) Platform Libraries:
These include Java and C/C++ libraries like Media, Graphics, OpenGL, etc. for android development.

5) Linux Kernel:
This is the heart of Android Architecture. It manages multiple drivers like display drivers, camera drivers, Bluetooth drivers, etc. which are required during runtime. It also provides an abstraction layer between the device hardware and other components of the android architecture. Linux Kernel also provides the main feature of security in android architecture.

III. Inbuilt Security Features in Android System

Android OS provides some inbuilt security features to make the device as secure as possible Some of these features can be listed as:

1) Application Sandbox:
   Android applications execute in a sandbox. Sandboxing is a technique where applications cannot access other system resources without explicit access permissions granted by the user during installation. [4]
2) Application Signing:
   Android requires that all apps be digitally signed with a certificate before they can be installed. Android uses this certificate to identify the author of an app, and the certificate does not need to be signed by a certificate authority. Android apps often use self-signed certificates. The app developer holds the certificate's private key. When the system installs an update to an application, it compares the certificate in the new version with those in the existing version and allows the update if the certificate matches. Every application that runs on the Android platform must be signed. [5]
3) Encryption:
   Once a device is encrypted, all user-created data is automatically encrypted before committing to disk and all reads automatically decrypt data before returning it to the calling process. It ensures that if a third party tries to access the data they won't be able to read it.

IV. Possible Threats/Risks in Android Security

The threats in android security can be classified according to the methods that are used to acquire access to the device and the methods that are used to attack the device.

1) Touchscreen and Buttons Data Integrity Violations:
   Fake data is emulated through Android Debug Bridge (ADB) shell and different operations can be performed on the device such as installing applications, debugging applications, accessing the Linux shell. It also provides remote access to a device either through a wired or a wireless connection. [3]
2) GPS Data Falsification:
   This method generates a false GPS location and it is done in two ways. The first way is by going to the developer menu and turning on the "Allow Mock Location" option and the second way is to "root" the phone and faking a location. Rooting the phone means gaining Super User privileges. [3]
3) Environment, Position and Motion Sensor Data Manipulation:
   This attack requires the device to be rooted. It changes/modifies the data collected from the sensors such as temperature sensor, illuminance sensor, magnetometer, etc. [3]

If the device of a user is rooted then a non-legitimate user can access multiple applications and read data illegally. This requires the access of permissions from the dangerous category and it can only access if the user has SuperUser rights i.e., the device is rooted. These dangerous permissions include [3]:

PERMISSION GROUPS	PERMISSIONS
Calendar	READ_CALENDAR WRITE_CALENDAR
Camera	CAMERA
Contacts	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS
Location	ACCESS_FINE_LOCATION ACCESS_COURSE_LOCATION
Microphone	RECORD_AUDIO
Phone	READ_PHONE_STATE CALL_PHONE READ_CALL_LOG WRITE_CALL_LOG ADD_VOICEMAIL USE_SIP PROCESS_OUTGOING_CALLS
Sensors	BODY_SENSORS
SMS	SEND_SMS RECEIVE_SMS READ_SMS

V. Metrics Involved in Android Security Evaluation

The security for an Android OS can be scored using certain metrics. These metrics are related to Android OS security and will help in eliminating the possible attacks through Dangerous Permissions. The metrics can be split into 2 different groups:

- Internal metrics: These metrics are related to the device data.

- External metrics: These metrics are related to external information.

A. Internal Metrics

(a) Root Access:
Achieving root access permits to fake any sensor data on a device. It can be achieved through flashing third-party Android OS images or using software such as Kingo Root to gain root access. The metric for root vulnerability has 2 values: 0 – the root access is gained and 1 – the root access is not gained. [3]

(b) Device Lock:
Android provides its user with different forms of device lock that include: password, pattern, and PIN. If none of these forms of device lock are implemented in your device, it becomes easier for the non-legitimate user to access the device. To help with that Android OS provides a secondary lock mechanism that makes it easier for the user. These include fingerprint, face scan, trusted places, trusted voices, and trusted devices. These mechanisms are less secure; hence the Android OS in most cases makes the user use one of the primary lock mechanisms while setting up the secondary one. The metric for lock mechanisms are as follows: 0 – no lock mechanism, 1 – secondary lock mechanism, and 2 – only primary lock mechanism.[3]

(c) Android OS Versions:
The latest versions of Android come with more security and advancements. If the device is still using an older version of Android OS, it increases the vulnerability of the device. The metric for this can be set as: 0 - the device has an outdated Android OS version, 1 – previous Android OS version, 2 – the latest android OS version.[3]

(d) Unlocked Bootloader:
Bootloader loads either OS or recovery software. If the bootloader is locked, Android OS will not allow loading unverified or untrusted third-party recovery software or Android OS. If the bootloader is unlocked, it provides "root access". The metrics for bootloader are as follows: 0 – bootloader is unlocked and 1 – bootloader is locked. [3]

(e) Security Patch Version:
A security patch is released by the application makers if they detect a vulnerability in the previous version. If the security patch is not up-to-date it will increase the vulnerability of the device i.e., having the latest version of security patch is most secure followed by a version older followed by an outdated version. The metric score can be shown as follows: 0 – outdated security patch, 1 – previous security patch, 2 – latest security patch. [3]

(f) Unknown Sources of Application:
An application not installed from Google Play might contain malicious data that will contaminate the data on the device since it will not have a signature verifying the contents of the application. This can affect the data as well as manipulate sensor data. The metrics for this can be shown as: 0 – if unknown source applications are allowed, 1 – if not. [3]

(g) Developer Options Menu:
If the developer options menu is enabled the user's sensor data can be manipulated such as GPS data. If the device has been rooted i.e., the root access is gained then it makes no difference if the user has enabled developer options or not. The metric for the same are as follows: 0 – if developer options are enabled, 1 – if not. [3]

B. External Metrics

(a) Installed Applications:
This metric takes into account several factors, firstly, it takes into account what applications are installed on the device and then checks if the signature on those applications is the same as the ones on Google Play Store. If the signatures match, then the applications can be considered safe, else they are installed through a third party and might contain malicious data. The metric for this will have a range from 0 – 3: 0 – a known application for data manipulation has been installed, 1 – an unknown third-party application has been installed, 2 – all applications are known but signatures are different from the ones on Google Play Store, 3 – all applications match the signature to the ones on Google Play Store. [3]

(b) System Vulnerabilities:
This metric takes into account 3 different factors – the Android OS version, a device model, and versions of the installed security patches. The version of all the three would decide the metric value: 0 – System has well-known vulnerabilities, 1 – the system does not have well-known vulnerabilities, but a device has inaccurate sensors, 2 – system has no vulnerabilities and has accurate sensors. [3]

(c) Device Rating:
This metric is different from others as it depends on the reliability of the data provided by the device. The metric system varies from 0 to 9: 0 – if a device provided fake data more than 8 times, 9 – if the device did not provide any fake data. [3]

VI. Implementation of Hierarchical Expert System

An expert system is a computer system that emulates the decision-making ability of a human expert to solve complex problems in a particular domain. It performs this by extracting knowledge from its knowledge base which is commonly represented in the form of IF-THEN type rules, using the reasoning and inference rules according to the user's queries. [6] Expert systems are considered to be of high performance, supposed to be understandable, reliable, and should be highly responsive. There are 3 main components of an Expert System: the User Interface, the Inference Engine, and the Knowledge Base.

1) User Interface:
The Expert System is able to interact with the user using the user interface. The ES takes in queries as an input in a readable format and sends it to the Inference Engine. After the input is processed and an output is generated by the inference engine it is displayed to the user using the interface.
2) Inference Engine:
This component acts as the brain of the ES since it is the main processing unit of the system. It takes the knowledge base and applies the inference rules to derive a conclusion or deduce some new information. The inference engine uses 2 different methods of deriving the solutions: Forward Chaining and Backward Chaining.
3) Knowledge Base:
This component stores the knowledge acquired from the different experts of the particular domain. The higher the knowledge in knowledge base, the more precise will be the ES.

Some of the main advantages of using an ES in the field of security are as follows:

1) Reduced Cost:
The development of an ES is relatively inexpensive. The repeated use by multiple organizations reduces the cost of the service per client greatly.
2) Increased Availability:
The data required for ES is available in abundance at this point of time. Web-based ES have the ability to access expertise from any internet connected device.
3) Multiple Expertise:
Due to the availability of abundance knowledge on internet through multiple sources, the total level of expertise of the system also keeps increasing in different domains.
4) Time Saving:
One of the main features of ES is that it provides high performance which reduces the amount of time taken to perform certain tasks and saves up quite some time.
5) Unemotional and Steady Response at all Times:
Unlike humans, the ES is a program and hence the human factor influence decreases.

The information security can be classified into different categories and these categories play a key role in organization's security assessment:

- Vulnerabilities: any weaknesses in the system of controls that might be exploited by threats.
- Threats: generally, people, things, or situations that could potentially cause loss.
- Assets: anything that has value to the system.
- Impacts: what would be the (worst case) effects if some of those threats materialized.
- Control: used to mitigate vulnerabilities by implementing either organizational or physical measures.[6]

All of the above categories should be taken into consideration to perform qualifies security estimation. Also, since the possibility of something happening, especially an Information Security event, is very hard to evaluate precisely, it is represented as fuzzy terms. The impact of vulnerability on the particular threat is reflected in rules, which have the following pattern:

```
IF vulnerability is very serious,
THEN threat execution possibility
is low/moderate/high (fuzzy value)
```

To see how the ES uses the described knowledge base to assess information security, the ES asks the user with a set of questions using its knowledge base, and evaluates the security through the following steps:

1) Collection of User Data:
All the knowledge about the assets is gathered and the following information is expected:
   - assets that are present

- values of the assets
- the dependencies between them

2) Threats, Vulnerability and Impacts Identification: After all necessary information is collected, the system tries to find vulnerabilities that can exploit these threats and all of its impacts.
3) To assess the quality of implementation, the ES asks several questions regarding each of standard controls:
   - Frequency of backups of sensitive data
4) Collecting Smartphone's Data:
   An application to collect smartphone's data is used. The data collected includes –
   - Device Control
   - Hardware Description
   - Onboard Sensors
   - Installed applications and permissions

The information generated by the application helps the expert system in evaluating the security of an Android OS. It also improves the knowledge base of the ES since new data might be derived from the existing data. The ES then finds the appropriate threat, vulnerability, control instances, impacts, and the new set of questions (in reference to the device) populating the knowledge base.

The attacks on the device are a result of the threats made through the various security vulnerabilities. The result of analysis of security vulnerabilities, properties, sources of threats such as nature of the occurrence, character and possible probabilities implementation in a particular environment provide us with information regarding given set of information resources which allows in defining the security policy. [7] This classification of sources of threats and their display may be carried out by analysis of interaction of logical chains which are constructed using the security policy and the analysis of possible risks. The logical chain can be represented as follows:

$$SourceofThreat \rightarrow Threat \rightarrow Vulnerability \rightarrow ImplementationofThreat \rightarrow Impact \rightarrow Controls$$

A. Classification of Threats

The sources of the threats can be classified as external sources and internal sources and both external and internal sources can be intentional and unintentional.

- Unintentional Threat:
  Arise regardless of the will and the desire of the people. It is mostly associated with direct natural or anthropogenic impacts on physical elements of the system.
- Intentional Threat:
  These are created only by people to disrupt the work of an information system. It can further be classified into two different threats – Passive threats, which are related to unauthorized access, and Active threats, which are related to the attempts to change the information of the legitimate user.[7]

B. Classification of Vulnerabilities

Vulnerabilities are present on both the hardware as well as the software and each threat can be compared with different vulnerabilities i.e.:

- Objective Vulnerability
- Subjective Vulnerability
- Occasional Vulnerability

C. Threats, Vulnerability and Control Scenario in Android Device

- Threats and Risk Analysis of an Android Device:
  - Classify and rank the files and applications to be protected in order of importance.
  - Identify potential threats to security and ways to implement them.
  - Assess the impact of these threats.
- The main files and applications that need to be protected include:
  - Banking Information
  - Personal Data including Images and Messages
  - Access to Camera, WiFi and Bluetooth

The threats can now be classified through the above description for the Android Device:

1) Unintentional Threat:
   TH1: Threat of user displaying the pin
   TH2: Downloading a third-party application that does not have a valid signature
   TH3: Giving permissions to third party applications with invalid signatures
2) Intentional Threats:
   TH1: Rooting the Android Device
   TH2: Installing applications from pirated sources
3) Vulnerabilities:
   V1: Root Access
   V2: Unlocked Bootloader
   V3: No Device Lock
   V4: Old Android OS version
   V5: Developer option enabled

4) Controls:
   C1: No Root Access (Device is not rooted)
   C2: Locked Bootloader
   C3: Device is locked using primary lock i.e., PIN/-Password
   C4: Prevent installation of applications that do not have a valid signature
   C5: Keep the device updated with latest OS version and Security patches
   C6: Developer option disabled

Through the above scenario, we can build a knowledge base of Threats (THi), Vulnerabilities (Vi), and Controls (Ci) for an expert system for Android OS. [2]

The user's answer will be taken and processed for the knowledge base for the expert system and later will be converted to numeric values by using their weights for each variant. These numeric values will be obtained using the metric system in the previous section and then the answers in the knowledge base will be used to train the expert system.

VII. Conclusion

It is clear that Android devices are one of the most used smartphones in the world and these devices hold information that can be very personal to a user. It holds crucial information like contacts, messages, bank details, health details, etc., while also having access to features like trading stocks, managing banking, accessing insurance, unlocking and locking the house, access to all the other electronics used by the user, and many more. With all these features the amount of threat for these devices also increases significantly and hence security becomes of great importance to all individuals and organizations related to the device. Hence for evaluating the security of the Android device we can develop an application (Project 2) that performs the evaluation on certain metrics:

1) Network Security Status:
   - WiFi Status
   - Airplane Mode Status
2) Application Security Status:
   - OS Difference Status
   - Encryption Status
   - Root Access Status
3) Device Security Status:
   - Screen Lock Status
4) Software Security Status:
   - Application Signature Status
   - High Risk Application Status

Metric	Scale
WiFi Access	Boolean
Airplane Mode	Boolean
OS Version	From 0 to 2
Device Lock	Boolean
Root Access	Boolean
Developer Options	Boolean
Installed Applications Rating	From 0 to 9

The main idea is to build an expert shell using the rules above. As the application is downloaded, it will access and try to learn and assign metric values depending on the Android system and will provide a score that will describe the security of the device. If certain data cannot be accessed the user will be asked to choose whether their device has the following feature enabled or disabled which will again add to the final score of security evaluation. The application will be built on Android Studio and hence will be coded in Java and since Java is being used the application might use JESS as the expert system. For project 3, the data used to calculate the security evaluation i.e., the metrics will be used to develop a machine learning model for approximation of the ES input-output surface. The idea is to build a machine learning model in Python libraries: Pandas and Sklearn and TensorFlow. The main software requirement for the same is going to be only PyCharm and Python 3.0.

References

[1] (2020) Mobile operating system market share worldwide. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide
[2] N. Raza and N. Kirit, "Security evaluation for android os using expert systems," June 2019.
[3] I. Khokhlov and L. Reznik, "Data security evaluation for mobile android devices," 20th Conference of Open Innovations Association (FRUCT), 2017.
[4] J. Khan and S. Shahzad, "Android architecture and related security risks," 2015.
[5] S. Chatterjee, K. Paul, R. Roy, and AsokeNath, "A comprehensive study on security issues in android mobile phone – scope and challenges," 2016.
[6] K. Kozhakhmet, G. Bortsova, A. Inoue, and L. Atymtayeva, "Expert system for security audit using fuzzy logic," CEUR Workshop Proceedings, 2012.
[7] L. Atymtayeva, K. Kozhakhmet, and G. Bortsova, "Building a knowledge base for expert system in information security," 2014.