FOUNDATIONS OF INTELLIGENT SYSTEMS

PROJECT 2

Submitted to:
Prof. Leonid Reznik

By:
Shivang Bokolia

Executive Summary

Smartphone security has become one of the most important features of a smartphone and has also become a challenge for different organizations and individuals. Smartphone store crucial information of a user like contacts, messages, bank details, etc. while also having access to features like stock trading, accessing insurance, locking and unlocking housing and automobiles and many more. With all this information the amount of threat on these devices increases as well and hence security of the device becomes of utmost importance.

The goal of this project is to design a simple yet effective application that will evaluate the security of the smartphone based on different data collected from the device. This application will make use of a hierarchical expert system and will provide the user with a total score and description about the security of the smartphone. The application will be specific to Android Smartphones only.

Requirements

The project required multiple phases which included:

1. Understanding the different features and data that could be used to break the security of the device.
2. All the data from above was taken and a bunch of facts were prepared. These facts were used to build the knowledge base.
3. Several rules were formulated to act on the knowledge base to either produce new facts or verify the existing ones.
4. The last part included developing the expert shell with all the facts and rules and running it in the application to evaluate the security of the system.

Specifications

The project consists of the following sections:

1. Read the current settings and data of the device:
   The device for this project only involves Android smartphones and all the data and information that I gathered, is done for the Android Smartphones only. There is a large amount of data and features available in an android smartphone, but the ones that are relevant to this project are related to security only. This includes – OS Version, Developer Option Settings, Verification of applications, Third-party applications, etc. All of this data is recorded from the device in real time.

2. Determining Facts and formulating rules:
   The main objective is to take all the data and information related to security of the device from the recorded data. All the information obtained are set as facts about the device's

security and are stored in the knowledge base. This data includes different features of an Android Smartphone and hence cannot be accessed at the same time and hence need to be grouped into relevant categories. This includes – OS Version and Developer Option Settings as the System Security Evaluation, verification of applications and third-party applications as the Application Security Evaluation and others.
All the basic facts like the OS version, verification of applications, etc. are stored in the knowledge base and other facts like System Security Evaluation, Application Security Evaluation, etc. are derived using a set of rules which are formulated according to the device's security requirement. As new facts are obtained, the knowledge base of the expert system is updated as well.

3. Expert Shell for executing the rules:
   The rules that are formulated from the facts and knowledge base are executed in the expert shell using IF – THEN statements. This makes the execution much faster and works well for the android application and also the results generated can be verified easily. The expert shell will access the rules and will generate facts or verify existing ones and the information generated will access the knowledge base. After all the rules are done, the final score of the security of the Android Device is generated with a description and is provided to the user.

4. User's POV:
   The user will open the application and will press the Start button to start the security evaluation. The security evaluation will begin and all the data will be collected and the facts and rules will be generated. The user will be asked to add data on their own as well and after the user is done, they can press the Check button to receive a security evaluation of their device along with the description.

5. Description:
   The user is allowed to click or press on any of the parameters that are involved in checking the security of the device. The user will be taken to the description page where they will be explained the score system for that certain parameter. They are allowed to go back and continue their testing from there.

6. Score System:
   The score for each of the factors in the application is provided as below:
   a. OS Version – 0 if latest OS Version, 1 if previous OS Version and 2 if outdated OS Version.
   b. Developer's Options – 0 if the developer's options are enabled, 1 if disabled.
   c. Wi-Fi – 0 if Wi-Fi is not connected or turned off, 1 if connected to a network.
   d. Verified Applications – 0 if non-Google PlayStore apps are not allowed on the device, 1 if they are allowed.
   e. Device Lock – 0 if there is a lock on the device, 1 if not.

f. Third-Party Applications – 0 if the user has not checked the box, 1 if the user checked the box.

Description of the domain problem

The expert system that is implemented in the project follows the Forward-Chaining process. The expert shell has been created from scratch depending on the data that is obtained from the user's android device. Since the project was developed for Android Systems and devices, Android Studio was used to develop this application and all the code written was done in Java.

The expert system was created manually because no other existing Expert System Libraries or Languages are compatible with Android Studio. The original idea was to use JESS since it is an expert system that can be used in Java, but it is not compatible with Android Studio and hence was not used. CLIPs was another option that could be used as an expert system in Android Studio but that would require extensive knowledge of using Android Native Development Kit (NDK) and due to the time constraint, it was not possible.

The expert shell in the application is configured in such a way that it collects the data from the device and generates different facts. The rules are then formulated and act upon the current knowledge base to generate more facts. These facts generate more rules and provide user with a better solution. Since the facts are all available and the final result is not, the best approach was to use Forward-Chaining.

Implementation

Knowledge Base and Database:

Database:
For the storing the facts generated, we use a Hashset. All the facts that are generated after the collection of device's data are stored in the Hashset and since a hashset only stores unique data, we don't have to worry about repetition of the facts. Also since the hashset has a access time of O(1) it becomes much faster for the system as well.

Knowledge Base:
The knowledge base is populated with facts as the data from the user's device is being collected. There are different IF – THEN statements that produce a structure as displayed below and generates a score for the device's security.

OS_Version ^ Developer_Options → Software_Security (SS)

Verification_of_Application ^ Third-party_Applications → Application_Security (AS)

Device_Lock → Device_Security (DS)

WiFi_Settings → Network_Security (NS)

SS → 1
AS → 1
DS → 1
NS → 1

SS ^ AS → 2
SS ^ DS → 2
SS ^ NS → 2
AS ^ DS → 2
AS ^ NS → 2
DS ^ NS → 2

SS ^ AS ^ DS → 3
SS ^ AS ^ NS → 3
SS ^ DS ^ NS → 3
AS ^ DS ^ NS → 3

SS ^ AS ^ DS ^ NS → 4

SS – System Security
AS – Application Security
DS – Device Security
NS – Network Security

Description of the expert system application

The application for this project is heavily dependent on the expert system for providing the correct results. There are several other applications where this expert system can be used. These applications include:

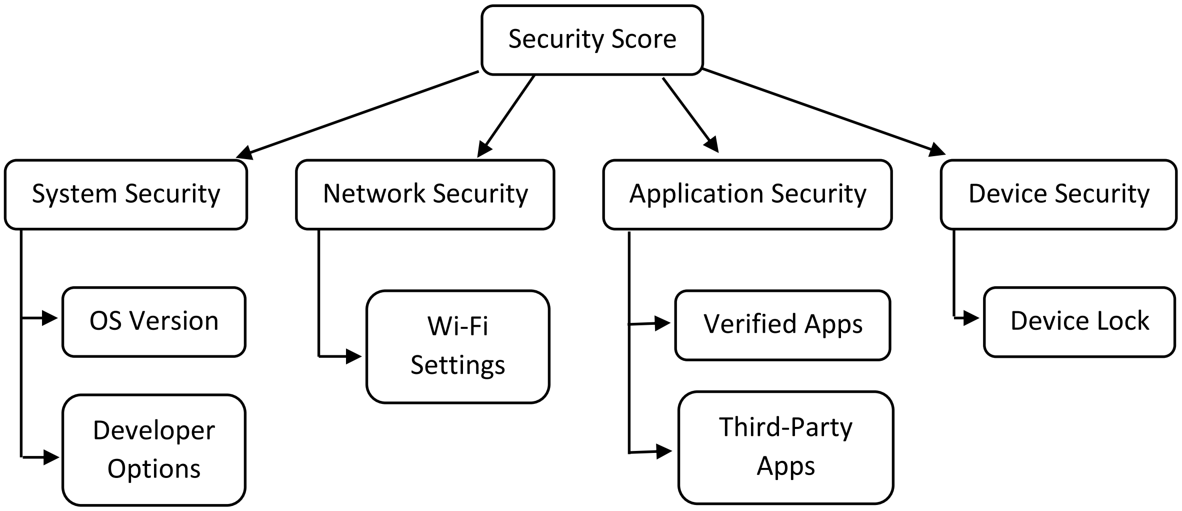
1. Investment Manager:
   Investing in stocks require certain rules. These rules can help in increasing your performance in the market. These rules can be applied in an expert system and using these rules the expert system can provide information on when to buy or sell which stock.

2. Diet Planning:
   There are rules involved when it comes to dieting. These rules include the number of calories, amount of exercise, etc. These rules can be implemented in an expert system which will provide the user with a dietary plan for them.

3. Better Applications:
   The current application can be used to provide the level of security for the user for their device. Some more rules and facts in the knowledge base with a good database can help provide the user with better applications compared to their harmful ones.

Structure

The expert system implemented in the application follows a hierarchical structure. This structure can be shown as:



Content

Data from device	Score	Information
OS Version	0	Latest OS Version
	1	Previous OS Version
	2	Outdated OS Version
Developer Options	0	Developer Options Disabled
	1	Developer Options Enabled
Wi-Fi Settings	0	Wi-Fi is not connected
	1	Wi-Fi connection is not safe
Verification of Apps	0	Apps are verified (Play Store)
	1	Apps are not verified (Play Store)
Third-Party Apps	0	No Third-Party Apps
	1	Third-Party Apps installed
Device Lock	0	Device is Locked
	1	Device is not Locked

User Interface

The user interface of the application is very simple and easy to understand for the user. The user opens the application and presses the Start Button to begin the security evaluation of the device. After pressing the start button, they are taken to the second screen, where the data for each security factor is shown. The user is asked to press the Check Button to check the final security score of the device. On the final screen the security score of the security of the device is provided along with a small description about the risk on the device.

Limitations

The application is to check only the top features of the device. If there are system vulnerabilities deep inside the system, the application will not be able to check those and those vulnerabilities will not be brought to the user’s attention through this application.

The second limitation is checking for all the applications. Due to constraint in time and knowledge of Android Studio, it was not possible to check all the applications in the device for any third-party applications. Hence the user is asked to check a checkbox if they know of any third-party applications involved in their device. Also checking all the applications in the device would make the device very slow which is not what we want.

Software Requirements

The application is only meant to run on Android System or Android devices. The android devices should have an API system version above 20.0 to get all the factors checked.

Hardware Requirements

All of the Android Smartphones can be considered for this application. There are no specific requirements for the applications.

Classes Used

There are 4 different classes involved in the project:

1. MainActivity Class:
   The MainActivity class is where the application is executed from. This holds the first page of the application with the Start Button. This class is where all the data from the device of the user is extracted and the facts for the knowledge base are generated.

2. Evaluation Class:
   The Evaluation class is where the Expert Shell for the application exists. All the rules of the expert shell are implemented here and new facts are generated and added to the knowledge base here. After all the rules are executed in this class, the final score for the security of user's device is calculated here as well using the expert shell. This class then sends the data to the FinalScore Class for display.

3. FinalScore Class:
   This class displays the final score of the security of the user's device. It also provides the user with the risk evaluation of the device.

4. Description Class:
   This class holds the description for all the factors and information that were retrieved from the device. It provides an explanation for the score that was provided in the Evaluation class for all the information.

The application consists of normal classes used to build an application. The specific classes used for retrieving data in the device that were imported have been mentioned below:

```
import android.app.KeyguardManager;
import android.content.Context;
import android.net.ConnectivityManager;
import android.os.Build;
import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.TextView;
```

- import android.app.KeyguardManager:
  This class is used for checking if the device is locked or not.

- import android.net.ConnectivityManager;
  This class is used to check the network status of the device.

- The main class that operates in the application is the "MainActivity" class. This class consists of all the logic behind the application as well as the expert shell for the expert system in the application.

Testing Description

There were multiple tests performed to check if the application was working the way it is supposed to. It was checked whether all the parameters that need to be generated on using the applications were done the way they were expected to –

1. Screen Lock – The device lock parameter was tested by removing the device lock first and checking whether the application gave a value ‘1’, then the device was locked with a PIN and it was checked whether the application is providing the value ‘0’.

2. Developer’s Options – The developer’s options were disabled at the beginning and it was checked whether the device provided the value of ‘0’ and later were enabled and the application was run again to check whether the value ‘1’ was received this time.

3. OS Version – An android virtual device was run for this testing and the settings for this virtual device were made in such a way that it had an outdated OS and the application was expected to provide the value ‘2’. Then a new virtual device was used with the previous OS Version, to check if the application provided the value ‘1’. The application was then run on a device with the latest OS and the expected output was ‘0’.

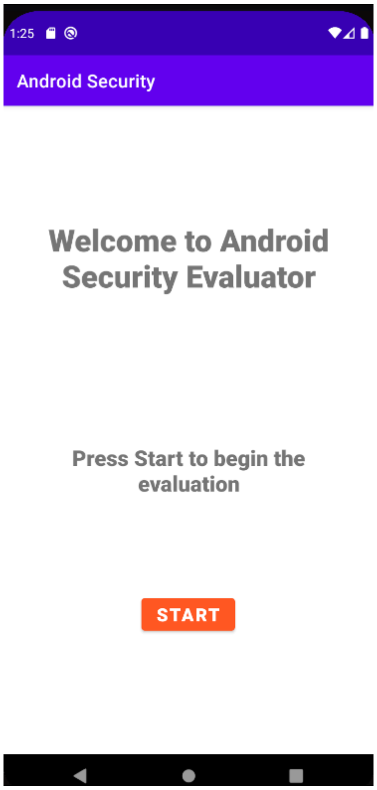
4. Network Check – The android device was first connected to a public Wi-Fi and the expected output from the application was ‘1’, and later the Wi-Fi was disabled in the device and the expected output for the application was ‘0’.

5. Third-Party Applications Check – For this, the user has been provided with a checkbox. The checkbox is supposed to be checked if the user has any third-party applications on their device and the application should provide an expected output of ‘1’. If the box is not checked then the expected output from the application is ‘0’.

6. UI Testing – For the UI Testing it was checked if all the texts and buttons were in the position they are expected to be in when the application is running. All the buttons in the application were pressed to check if they were working the way they are supposed to. The Start Button on the first page takes the user to the second page where all the data retrieved from the device is provided along with their security score. The Check button on the second page takes the user to the third page that provides the final security score of the device. All the texts on the second page were checked to provide the description and support page for that parameter.

GUI

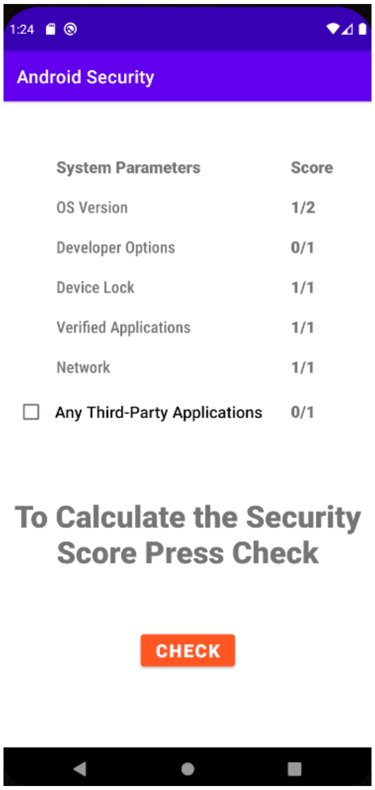
Main Home Page:
The main home page has the start button along with the description of the application. All the data of the user’s device is collected when the application is on this page. The start button will take the user to the next page that will have all the data retrieved and the score of all the data.



Welcome to Android Security Evaluator

Press Start to begin the evaluation

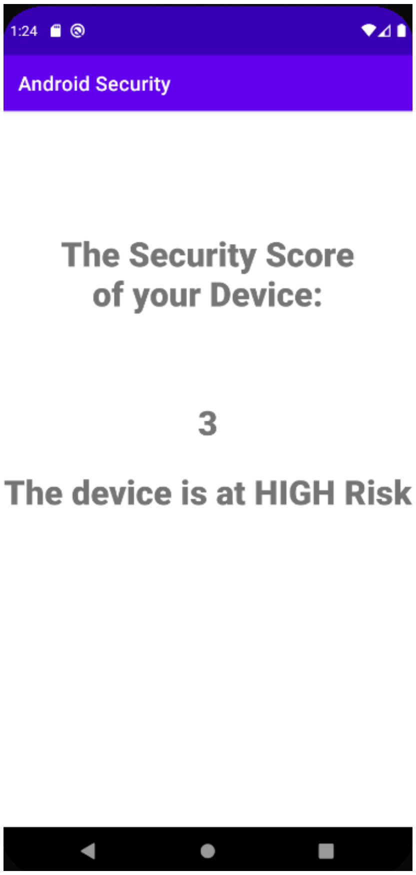
Data Retrieved and Score Page:
This page includes all the data and their security score that was retrieved from the user’s device. All the score generated is displayed to the user. This page also includes the Check button that will run the expert shell to provide the final score and description of the user’s device’s security.



System Parameters	Score
OS Version	1/2
Developer Options	0/1
Device Lock	1/1
Verified Applications	1/1
Network	1/1
□ Any Third-Party Applications	0/1

To Calculate the Security Score Press Check

Final Score Page:
This page will have the final security score of the device along with a small description of final risk level of the device.

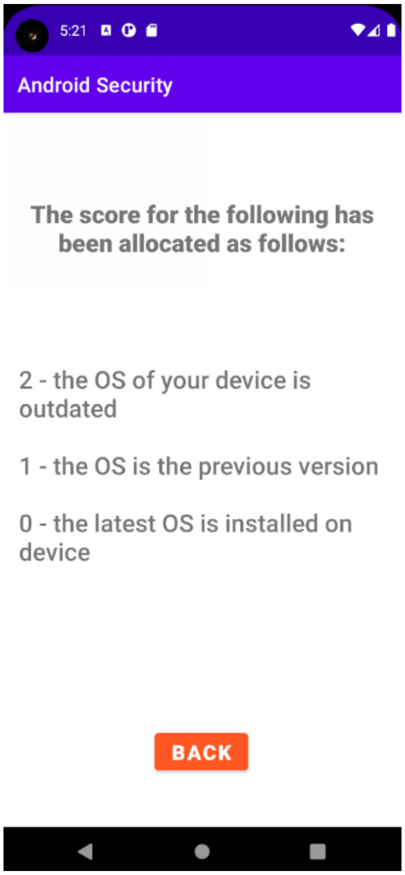


The Security Score of your Device:

3

The device is at HIGH Risk

Description Page:
This page consists of all the descriptions of all the system parameters that are involved in the security evaluation. This page can be reached by clicking on any of the parameters and will provide info about the score for the same.



The score for the following has been allocated as follows:

2 - the OS of your device is outdated

1 - the OS is the previous version

0 - the latest OS is installed on device

How to run the application

There are two ways to set up the application:

1. Install the application with the APK file:
    a. Download the application APK file.
    b. Open the file and install the application.

2. Run Code through Android Studio virtual phone device:
    a. Install Android Studio from https://developer.android.com/studio. Follow the instructions as per the operating system.
    b. Open Android Studio.
    c. Unzip the project in your preferred location.
    d. Load the project in Android Studio.
    e. Once the project is loaded run the project.
    f. Android Studio will ask for a virtual device to be used. Choose a virtual device with OS version above 25.
    g. As soon as the device is selected, the application will run automatically.