Final Year Project Plan

Full Unit – Project Plan

DEVELOP A SECURITY SUITE FOR ANDROID BASED SMARTPHONES.

Abd ElRahman Mohamed Hassan Abdou M Soliman

Supervisor: Christian Weinert

Department of Computer Science
Royal Holloway, University of London

October 07, 2022

1.1 Abstract

Since the first release of the Android mobile Operating System in 2008, Android Smartphones have developed to become an integral part of day-to-day life and as smartphones have improved and added features, the amount of personal/private data they handled also increased. Android Smartphones now have the capability to act as anything from a way to digitally sign contracts to complete bank branches in our pocket. They can replace anything from our Debit Cards to our personal laptops and as we become more and more reliant on the palm-sized slabs of glass in our pockets, the need for security becomes more and more crucial. Especially now more than ever as new malware is being developed and exploited quicker than exploits can be patched and as our devices become more and more connected to the outside world an up-to-date fully featured android security suite will help keep our phones secure.

To combat this, various Android smartphones manufacturers have been guaranteeing Android Security Updates on a monthly, bi-monthly, or quarterly basis for a limited number of years depending on the age of the device. This is beneficial in practice until the variation in number of years being supported from manufacturer to manufacturer is realised. For example, Google, which has 0.5% market share (appbrain.com, 2022), can promise at least 3 years of Security updates across all their with their latest smartphones offering at least 5 years (Google Support, 2022) whereas Oppo, with 10.1% market share (appbrain.com, 2022), uses a "more you spend, the more you get policy"(Android Authority, 2022) with mid-range and budget models receiving less and less Security updates as you go down in price with their ultra-budget phones seeing no updates. Maintaining the security and increasing the longevity of these phones is one of my motivations for this project.

For most people, Android already comes with all the essential safety and security features such as Encryption and App Security and basic malware and anti-virus protection. However, the level of security is dependent on whether the device is running the latest version of the OS. Unfortunately, the Android Version market Share is very fragmented with only 23.5% of devices running the latest Android 12.0 with more than 27% using a version of Android that is over 3 years (Android Version Market Share Worldwide | Statcounter Global Stats, 2022). Naturally, owners of older android devices look to the Google Play store for a 3rd party Security solution such as those provided by McAfee or Malwarebytes and then soon realise that in order to maintain every aspect of security on their device, they would need to download multiple apps because there is no full-featured Security suite that can act as an all-in-one solution for your security needs on your android device. This inconvenience may deter less tech-savvy users from maintaining the security of their older device. Having an all-in-one Security Suite on Android will benefit and allow those users to protect their data more conveniently.

I have taken an interest in this project as I am hoping to pursue a career in the Cybersecurity Industry with a particular interest in Mobile Security. I am an avid Android User who has resisted the rise of IOS for many years and have taken an interest in Secure Messaging and Encryption having programmed a secure messaging app (reminiscent of something like WhatsApp or Messenger) using Java and Google Firebase. Although I'm new to on-device android security, it has always been a subject that I have wanted to pursue in the future as I do believe that the Play Store's App Requirements are a bit relaxed in the security department and that this is a major issue as Android apps keep popping up on the news because of malware detection.

1.2 Goals and Objectives

With this project, my goal is to unite and package multiple existing android security modules into a do it all mobile security centre that lives on the user's android device of choice and conveniently shows the user an overview of the level of security that their device currently possesses in a user-friendly understandable way. I aim to achieve this by using open-source implementations of various security features as the foundation that will allow me to develop my own fit for purpose implementation that can be incorporated into my security suite without requiring a standalone app for each module or modifications to the OS. I aim to present the data being shown by my security suite in an understandable way that doesn't panic the user if not necessary yet will also allow the user to use multiple of the modules listed below with the press of a button in order to keep their security up to standards. A secondary goal is to then have this security suite run automatically in the background ensuring that all security definitions are up to date for the malware detection algorithm as well as provide day-to-day security without user input.

This project will analyse the possibility of having an all-in-one security suite on android that:

- Requires no prior modification to the OS
- User-Friendly
- Up to date using the latest security definitions
- Is Compatible with older versions of Android
- Handles multiple modules of Security on an Android Device from this list:
    - **Anti-Virus/Malware Detection**
    - Firewall
    - **Overview/Management of app permissions**
    - **Overview/Management of active sensor usage (including camera, microphone, etc.)**
    - **File Encryption (.zip, .rar encryption?)**
    - Password Manager
    - VPN/ Tor
    - Secure Messaging
    - **App Access Control**

2 Timeline

2.1 Term 1 (12 Weeks)

Preferably, I will spend Term 1 acquiring the required knowledge and coding skillset needed to complete my project and write a report on the different security apps/modules I could consider integrating into the security Suite. By the end of Term 1, I hope to have finalised the list of security modules I will be focusing on implementing as well as have a basic proof of concept application that implements a couple of modules to test the feasibility of a security suite.

Early Deliverables:

- A report describing the different security apps for integrating into the security suite.
- A Proof-of-Concept implementation of the security suite
- Test Proof-Of-Concept

Week	Goal
Week 1 - 3	• Research into Malware Detection with Machine Learning on Android including dodgy app permissions detection and sensor usage. • If in position too, Create Proof-Of-Concept
Week 3-4	• Research into modifying App Permissions on android on-app using administrator permissions • Research into viewing currently used sensors and what app is using them. • Create Proof-of-Concept. • Research into App Access Control on Android
Week 4-5	• Create simple app that can request administrator permissions and change app Permissions • Begin Designing Refined unified UI for the app
Week 5-7	• App Access Control proof of Concept app that uses current lock screen to secure apps • Test proof-of concept for Malware Detection
Week 6-7	• Final testing for Proof of Concepts and select which one to use in the December presentation • Ensure chosen proof of concept is tested and working with more tests
Week 7-9	• Heavily refine selected working Proof-Of-Concept into an app which will include my in progress unified UI design that I will be using for my project.
Week 9-10	• Complete Report describing different security apps using research collected from Proof of Concepts
Week 11 - 12	• Fine tune Proof of concept so it runs on an actual android device • Prepare for Interim Report and Demo • Write complete Bibliography of all sources used. (Section of Final Project)

2.2 Term 2 (11 Weeks)

Term 2 is when I will use my proof-of-concept that I created at the end of term 1 as the foundation to the final app and will mainly be focused on implementing all the modules I have chosen to implement an polishing the UI and getting the security suite publication ready with the intention of having an app that is polished enough to be launched on android along with a comprehensive user manual that can be released with it. Hopefully, most of my security modules will pass the proof-of -concept stage and be in a stage where they are mostly implementable in my final Security Suite and would only require minor to moderate fine tuning to be in a deployable state.

Final Deliverables:

- Design and Develop the Final Security Suite
- Complete report which must include a User Manual

Week 1	• Design User-Friendly User interface • (Towards end of week 1) Finalise the User Design and overall Design Language of the security suite • Write final project Section (motivations and aims)
Week 2	• Create the User Interface in Android Studio • Write User Manual for Setup and navigating the Main Menu (Write Final Project How to run description) • Write Final project Section 2 (Professional Issues)
Week 3-9	• Implement Each Security module into the security suite • Test Each module as is implemented then test app (ensure all modules play well together) • Write the How-To guide for each security module and update setup guide if necessary (Roughly 1 Week Per Module) • Write Section on Theory and Software Engineering method in Final Report
Week 10-11	• Record Project Demo for Final Project • Prepare for Project Demonstration

3 Risks and Mitigations

My project is not immune to risks that could potentially hinder my progress severely as although I believe my project is doable within the time frame, the project is ambitious, and any roadblocks could severely swerve my timeline off track. Below I will be discussing risks that I may encounter during my project as well as mitigations that will help me avoid the risks being a reality throughout my project.

3.1 Completing Tasks in the allotted time according to Project Plan

Important risk to mitigate for the sake of the project

Since I have a limited number of weeks to complete my project, it is crucial that I stick to the timeline for each task that I set for myself and agreed upon with my supervisor. Any significant deviation from this timeline might stall my progress and potentially reduce the quality of future tasks as I rush more to get back on track. To mitigate this, I will be reviewing the time I spend on each module of the suite regularly as I research more and understand the size of coding needed for each module to rebalance my timeline.

3.2 Getting lost in Version Control

Crucial risk to avoid and Mitigations are easy to uphold.

As I am planning to program each module of my Security Suite in parallel (Separate Android Studio projects) then merge them into one suite to ensure that I can switch my focus effortlessly between modules and continue working on the project when I encounter a bug that Is taking too long to fix. I will end up with multiple codebases each with their separate APK's and folder structure. To continue working efficiently with this system, I am going to ensure I am using a sensible and useable folder structure that will ensure that I know exactly what I am working on. On Gitlab, this translates to multiple branches for each module to follow the correct version control conventions and ensure my code is always up to date with what is in my repo.

3.3 Overhead Machine Learning in Malware Knowledge

Important risk to mitigate

I understand that to create an effective Malware Detector for Android that I would need to implement Machine Learning in some capacity. This is almost crucial for a successful Detector on Android as there is so many apps on the play store that may have malicious intent and it is not feasible to produce a list of all potential apps in play store and then update it as new apps arrive. Machine Learning will allow me to tell the security suite what to look for inside an Android app and apply this to all apps whether they were released a year ago or today. I see this as a potential issue as although I can understand what red flags a detector would need to look out for, I do not have much experience in Machine Learning. Should my knowledge in this area become an issue, I will consult with contacts in the Information Security Department to find any materials that could help me. If this does not work, then I will be looking for alternatives throughout the whole process and I can begin assessing the feasibility and switch to an alternative solution.

3.4 Android Backward Compatibility

Certain to be a problem but Mitigations are easy to uphold

One of the main motivations for my project is to ensure that the security suite also works on older android devices to keep devices that have stopped receiving updates secure. For me to test the

compatibility of my security suite, I need to be able to test across multiple versions of android (8.0 and above). However, all the android devices that I possess are running the latest version of android (13.0) meaning it would be difficult to test compatibility on physical devices. Thankfully, Android Studio allows me to run an Android Emulator where I can install and run multiple older versions of android and test my app on these virtual devices. I will also ask around my friends and family for any older android phones that I could use to still get some tests on a physical device.

3.5 Licensing Issues and Using Open-Source Projects

Important to look out for and Likely to be an issue

As I understand that I will not be able to create and deploy every security module in the timeframe allocated, I will be using Open-Source projects to advance the development of my project. This meaning that I would try to use Open-Source projects with the correct licensing that allows free for educational use within my app. Ensuring the correct use of other people's code and licensing is a crucial professional issue in the world of Computer Science. If I do come across an open-source project that does not possess the correct licensing that allows me to use it, I will either, if it's important to my project, attempt to contact the author of the project for permission in order to use their project or, and most likely, attempt to find a similar project that has the correct license or look towards creating my own implementation for the particular security module.

3.6 Using Potential Malware to test my Security Suite

Important and almost certain to be an issue however mitigations are easy to uphold

To test my security suite (specifically Malware Detection), I would need to 'sabotage' the device I am testing the security suite on with something that is able to trigger the malware detector. Whether that be actual malware or something that can trigger a false positive. As this poses a security threat to any of my personal devices, I must be very careful as to what device to test this on and whether any potential damage is reversable. To this effect, I am mitigating this risk by only testing malware on the app through a Virtual Device ran by Android Studio. This means that if something were to go wrong, I can just wipe and delete the virtual device and create a new one. This minimizes the risk to my personal devices and personal data.

4 Bibliography

Android Security Research (2017) An introduction to Android application security testing (by Nikolay Elenkov). [Online Video] Available at: https://www.youtube.com/watch?v=hRuNHUwiQJA

Goes through essential tools needed for analysing Android application like network traffic analysis which could be useful when creating the Malware Detection Aspect of my suite

Android Security Research (2017) Android security architecture (by Nikolay Elenkov). [Online Video] Available at: https://www.youtube.com/watch?v=3asW-nBU-JU

Gives an overview of the Android Security Architecture and talks about the major android subsystems and components that relate to security including permissions and device policy. This will be useful in general for my project as I need to understand these components and how to make them work with my project.

Elenkov, N. (2015) Android security internals an in-depth guide to Android's security architecture. [Book] San Francisco, CA: No Starch Press.

A more in-depth overview made by same Speaker of video below that should complement the video in helping me understand the Android Security Architecture.

Jiang, X. et al. (2013) Android Malware. [Book] (SpringerBriefs in computer science).

Goes through a general survey of Android Malware that covers topics like malware behaviour and classification in Android. Will be useful for my Malware Detection.

wwt.com. 2022. Test-Driven Development with Android. [online] Available at: <https://www.wwt.com/article/test-driven-development-with-android> [Accessed 6 October 2022].

Takes me through how to use TDD (Test-Driven Development) when developing an android app which will help me follow TDD and ensure that I am continuously testing my app and making sure where any potential errors are coming from and fix them.

Android Open-Source Project. 2022. *Android Security | Android Open-Source Project*. [online] Available at: <https://source.android.com/docs/security> [Accessed 6 October 2022].

Android Documentation for Android Security. Relevant as it helps me understand how Security works on Android

Medium. 2022. *The Layers of the Android Security Model*. [online] Available at: <https://proandroiddev.com/the-layers-of-the-android-security-model-90f471015ae6> [Accessed 6 October 2022].

Article going through the basics of the Android Security Model. Need to understand for my project.

Senanayake, J., Kalutarage, H. and Al-Kadri, M.O. (2021) "Android mobile malware detection using machine learning: A systematic review," Electronics (Basel), 10(13), p. 1606.

A Review of Malware detection using machine learning on android. Talks about different types of Malware analysis as well as the 2 ways malware detection can be performed on android. (Signature-based detection and behaviour-based detection). Relevant to how I proceed regarding my malware detection algorithm.

Mahindru, A. and Sangal, A.L. (2020) “MLDroid—framework for Android malware detection using machine learning techniques,” Neural computing & applications, 33(10), pp. 5183–5240.

Presents MLDroid which is a web-based framework that helps to detect malware from android apps using dynamic analysis.