|  |
| --- |
| Iowa State university |
| Android Security |
| Senior Design Project |
|  |
| **Alex Frisvold, Alex Meyer, Nazmus Sakib, Eric Van Buren** |
| **9/27/2011** |

|  |
| --- |
| Advisors: George Amaricai of Iowa State and Victor Lukasik of Boeing |

Contents

[**1.1 Problem/Need Statement 2**](#_Toc304860271)

[**1.2 Concept Sketch 3**](#_Toc304860272)

[**1.3 System Block Diagram 3**](#_Toc304860273)

[**1.4 Operating Environment**](#_Toc304860274) **4**

[**1.5 Expected End Product 4**](#_Toc304860275)

[**1.6 User Interface Description 4**](#_Toc304860276)

[**1.7 Functional Requirements**](#_Toc304860277) **5**

[**1.8 Non-Functional Requirements 5**](#_Toc304860278)

[**2. Work Plan 5**](#_Toc304860279)

[**2.1 Work Breakdown Structure 5**](#_Toc304860280)

[**2.2 Resource Requirements 5**](#_Toc304860281)

[**2.2a Time Commitments 5**](#_Toc304860282)

[**2.2b Financial Resources**](#_Toc304860283) **6**

[**2.2c Documentation Resources**](#_Toc304860284) **6**

[**2.2d Other Resources 6**](#_Toc304860285)

[**2.3 Project Schedule 6**](#_Toc304860286)

**2.3a** **Acquiring Documentation, Initial Research, and Setup 6**

[**2.3b Digging Deeper – A Complete Understanding**](#_Toc304860287) **7**

[**2.3c Implementation**](#_Toc304860288) **8**

[**2.3d Testing and Maintenance**](#_Toc304860289) **8**

[**2.3e Beyond**](#_Toc304860290) **9**

[**3 Risks and Risk Management**](#_Toc304860291) **9**

[**3.1 Not Possible to Implement**](#_Toc304860292) **9**

[**3.2 Time Constraints**](#_Toc304860293) **9**

[**3.3 Resources and Documentation**](#_Toc304860294) **9**

[**3.4 Other Risks**](#_Toc304860295) **9**

**1. Introduction**

This project plan will outline the requirements and expectations of the ISU Engineering College and Boeing for the development of an Android emulator that replicates TrustZone instruction execution. This project plan will change throughout the course of this project and will emphasize challenges that arise as well as solutions to those problems.

## 1.1 Problem/Need Statement

This document will detail the current plan of implementation for our senior design project. Our project is to develop a working emulator for an Android device, be it a cell phone or tablet, such that it is able to accurately represent an ARM TrustZone. Using this TrustZone we would like to implement a Trusted Platform Module (TPM) service. The services we are most interested in are a random number generator and a public/private key generator using the previously mentioned random number generator.

As of right now there are no commercial Android emulators that can correctly depict the workings of an ARM TrustZone. Therefore application developers are forced to test their devices on actual hardware to see if their code works. This is both dangerous to the device and time consuming. The logical next step is to try and emulate the inner workings of this hardware so that developers can then start writing applications to use it.

**1.1 Intended Audience**

The intended audience of this this document will be the senior design team, the review board, our advisor, and our client The Boeing Company.

**1.2 Product Scope**

This emulator would be able to be used by anyone designing applications on an Android device that would like to add another layer of security to their application design.

## 1.2 Concept Sketch

This concept sketch shows how our software stack will operate. At the bottom most layer will be the QEMU emulator with TrustZone implemented, this will be our hardware emulator layer. Above that will be our kernel layer, this will start our Fiasco family of the software stack that will include the Fiasco microkernel and extend to include the L4Re Run-Time Environment and L4 Android layer. All of this will be the base for our Android application that will test the software stack for the correct use of the TrustZone.

1.3 System Block Diagram
We must take the existing QEMU emulator and extend it to mimic the results of hardware that natively implements TrustZones for secure instructions. This idea of changing worlds will grant the application, and therefore the user, an extra layer of security.


## **1.4 Operating Environment**

The extended Android QEMU emulator will allow for the full system emulation of the L4Android operating system on a virtual ARM CPU that implements the TrustZone architecture on an x86 Linux host machine. The existing QEMU source is written in C and any necessary modifications will also be written in that language. The Fiasco.OC microkernel and the L4Re runtime environment are implemented in C and C++ so we will be using those languages to make any changes to those two components. The Android applications we will develop will be written in Java using the Eclipse IDE.

## **1.5 Expected End Product**

The final deliverable for this project will be a functioning software stack that emulates the ARM TrustZone and allows Android applications to make use of the TrustZone. Specifically we will be testing the functionality and security of the RNG and key generator. Android applications that demonstrate this will also be part of the end product.

## **1.6 User Interface Description**

The modifications proposed in this project plan should not affect the GUI currently implemented as a part of the Android QEMU emulator. A screenshot of the emulator is given below (image from: [http://developer.android.com/guide/developing/devices/emulator.html):](http://developer.android.com/guide/developing/devices/emulator.html%29%3A)


##

## **1.7 Functional Requirements**

1. The modified Fiasco.OC microkernel will run seamlessly over Mr. Winter’s extended version of QEMU.
2. The modified L4Re runtime environment will run seamlessly over the modified Fiasco.OC microkernel.
3. The L4Android operating system will run seamlessly over the modified L4Re runtime environment.
4. Our developed Android security application will run on the L4Android operating system and will provide a specified TPM services.
5. Another Android application will be able to use the TPM services provided by the developed Android security application.
6. Modifications made to any of the various components of the software stack should not adversely affect any of the existing functionality of any of the components.

## **1.8 Non-Functional Requirements**

1. The modified software stack should run at a usable speed.
2. The modified software stack should be stable and run reliably.
3. Modifications to QEMU, Fiasco.OC and L4Re should be written in C and C++ programming language on a Linux platform.

# **2. Work Plan**

## **2.1 Work Breakdown Structure**

A collaborative approach will be taken among team members in this project. This approach will have the team working on the same tasks. The team will all work together during each phase of the project as opposed to having certain members assigned to certain phases (eg. Implementation goes to two members and testing goes to two other members).

## **2.2 Resource Requirements**

### **2.2a Time Commitments**

Each of the four-team members has their own constraint on the amount of time available to commit to the project, as they have other classes and/or jobs. Each member has agreed upon a 6-hour per week dedication to the project, which is a combined 24 hours per week. This is very flexible and will most likely change depending upon what stage of the project the team is currently on. The team will also meet once weekly as a group and with faculty advisor, George Amariucai. Bi-weekly meetings are also scheduled with our Boeing contact.

###

### **2.2b Financial Resources**

Financial resources for this project are minimal. QEMU, Fiasco.OC, L4Re, L4Android, Android SDK and Eclipse IDE are all open-source and were available to the team at no cost. As of now, the only required items for this project include a Linux box with all the aforementioned required software installed on it. An Android OS phone for testing the software stack on is also required however the team has already managed to check out an Android OS phone from the CSG at no cost.

### **2.2c Documentation Resources**

A very large amount of the time the team will spend on this project will be dedicated to researching. Understanding how the all the various software components work are all key for this project to be successful. Documentation on Android OS and the L4Android variant of the OS is readily available because they are open-source. To get documentation on ARM TrustZone, the team’s Boeing contact has referenced a few documents and the team has also signed up on ARM’s website which enables them access to such documentation. QEMU, Fiasco.OC, L4Re are also open-source and documentation can be found online. We are also looking into the papers written by Johannes Winter as he is the lead researcher in this field.

### **2.2d Other Resources**

Other possible resources that have been proposed include a Debian machine with a NX Machine client on it. This would allow us to only have to set up one machine with the required software stack as well as allow us to work from anywhere on or off campus. This will give us greater flexibility for work times as a group.

## **2.3 Project Schedule**

The team’s design methodology will follow the Spiral model. The steps of this process are determine objectives, identify and resolve risks, development and tests, and finally to plan the next iteration. We believe this will give us the best chance to succeed as it allows us to move the project forward even if we meet an unavoidable risk.

#### 2.3a Acquiring Documentation, Initial Research, and Setup

Gaining a basic understanding for the project and how each element in the project ties together is the first thing the team must do. This initial research includes how TrustZone, TPM, Android OS, and QEMU all work on a very basic level. Each team member will also set up a Linux partition on their machine and install the Android SDK that runs QEMU. This first phase of the project will take place between late September and mid-October.

#### 2.3ai TrustZone and Trusted Platform Module

The team’s Boeing contact has referenced a few documents relating to ARM’s TrustZone and TPM. Initial research and basic understanding of TrustZones and TPM are a priority in the early stages of the project.

#### 2.3aii The Android OS

Documentation for the Android OS is readily available and easily found. Initial research requires the team to gain a basic understanding of the structure of the Android OS.

#### 2.3aiii Google’s QEMU

QEMU is an open-source Android OS emulator. However, Google has its own version of QEMU that it uses and this version is included in the Android SDK. The team will acquire the needed amount of documentation on Google’s QEMU. During this phase, each team member will install the Android SDK on their Linux partition.

#### 2.3aiv Similar projects

Research online to find if there are any similar projects to the one our team is doing. If so, find anything useful that these other projects may offer.

### **2.3b Digging Deeper – A Complete Understanding**

In this phase of the project, the team will begin to dig deeper into how everything is implemented in the project. Having a complete understanding is the precursor to taking elements that already exist and augmenting them to create something of our own. This includes having a complete understanding of TrustZone and TPMs, Google’s QEMU, and the Android OS. If at this stage in the project we decide that our original goal is not practical, we will explore alternative ideas. Expected timeframe for this phase is expected to be from mid-October to mid-December.

#### 2.3bi Understanding Google’s QEMU source code

Understanding Google’s QEMU source code is a vital part of this project. The team estimates that this will be an ongoing goal of analyzing the code that already exists. The source code for QEMU is extensive and contains 273 files. Goals in this phase include analyzing both the official QEMU API and Google’s QEMU API and figuring out differences between the two.

#### 2.3bii Other Possible Scenarios

There is a possibility that after extensive research, this project may not be practical to do in what time we have. Boeing has informed the team that if a scenario like this arises then the team can make changes to the final goal.

##### 2.3biia How close to come to running software TPM on Android OS

If the team finds that it is not possible in the timeframe of this project to implement an emulator that can support TrustZone, then the team will explore what is required and how close one can come to running a software TPM on an Android OS.

##### 2.3biib A new goal with the information learned

Another option would be using the information the team has learned up to this point in the project and creating something else with it. Under the approval of Boeing, the team would explore other possible things to do with TrustZone, Android OS, and QEMU.

### **2.3c Implementation**

The implementation phase of the project will begin in the second semester of Senior Design. In this phase we will test each separate part of the software stack to ensure functionality as well as security and then bring the parts together to form our final software stack. Timeframe for this phase is expected to be from January to mid-March.

### **2.3d Testing and Maintenance**

The testing phase will run parallel with the implementation phase and also go beyond. Extensive testing of our code includes not only testing the augmented code on the stack but also testing code on the Android device. The team will be using forensics software to assure that the code is being properly emulated in the TrustZone. Expected timeframe of testing is from January to mid-April.

Maintenance may be required for the team’s software stack for many reasons. If updates in the Android SDK affect how the augmented QEMU code runs or if there are bugs that are discovered after the testing phase then the code will need to be fixed. The maintenance phase is expected to be from mid-April to May.

As of the time of this writing our team is in the process of getting a dedicated work server set up with Debian as the host operating system. We choose this OS because of the Linux environment and that it is recommended as the development platform for our Fiasco software stack. Keeping this project centralized will not only make working on it easier, but also make keeping track of changes easier. We are also in the process of setting up a Google code repository; this will allow us to keep track of changes made as well as store a copy of our work remotely.

Final testing of our working software stack will be done by implementing two TPM services, namely a random number generator and a public/private key generator. If we can create these inside the TrustZone without other applications being able to access them, this will be considered a success.

###

### **2.3e Beyond**

If a situation arises in which the project is far easier to implement than the team had originally expected then the team will explore going further with the project. This includes taking what the team has created and putting it to practical use. Possibilities may include creating applications that will show the forensics of what is going on and also penetration testing.

# **3 Risks and Risk Management**

## **3.1 Not Possible to Implement**

The ideal final goal of this project is to create a software stack that correctly emulates and uses the ARM TrustZone to produce a useable TPM that Android applications can use. After initial meetings with our Boeing contact, we were informed that this may not even be possible. From our research and what our Boeing contact has told us, this has never been done before. However, this is the ideal final goal, if this is not possible then exploring how close to running a software TPM on the Android OS would be our goal. Mitigation of this risk would be researching and trying to determine early on if it is possible or not to have an emulator that can support TrustZone.

## **3.2 Time Constraints**

If we find that it is possible to implement an emulator that can support TrustZone, we still face the risk of time constraints. We are unsure how to accurately predict how long it would take to successfully complete our ideal final goal of running code in a TrustZone of an Android OS emulator. The issue we run into is if it takes more time than we have to complete this project. Mitigation for this risk includes doing enough research early on and accurately gauging our time constraints.

## **3.3 Resources and Documentation**

Resources required for this project include an Android device, a machine with Linux installed, and Google’s QEMU. However, after some early research we are worried there may not be satisfactory documentation on Google’s version of QEMU to understand it enough to augment it. In order to combat this risk we will need to gather documentation on QEMU and come up with a solid understanding of how it works.

## **3.4 Other Risks**

If it comes up that this project seems that it may not be practical to do because of one of many reasons, Boeing has left us with the comfort of keeping this project open to changes. There would be no major changes; we would still be working with Android OS and probably also with TrustZone and QEMU.