DEPARTMENT OF COMPUTER SCIENCE
ROYAL HOLLOWAY, UNIVERSITY OF LONDON

Full Unit Project - Plan

IY3821 - Detecting Android Covert Channels

BSc (Hons) in Computer Science (Information Security)

Bell, Katrina (2017)
Supervisor: Jorge Blasco Alis
29/09/17

Abstract:

The project that I plan to complete over this year is ‘Detecting android covert channels’. Android covert channels are a form of communication between two applications on the same device that may be used for malicious reasons. Malicious intent is common in terms of covert channels, channelling private data from one app to the other in order to publish or send back to the attacker. While one app listens, the other sends data, for example, through an audio channel. Channels can be in many forms such as screen brightness controls, volume controls and resource files that are shared. Further there are two main types of covert channel, Storage channels and timing channels. The changing of a shared resource’s contents is a storage channel, while the changing of the timing of network activity is a timing channel (e.g. the delayed sending of packets) (Wade Gasior n.d.).

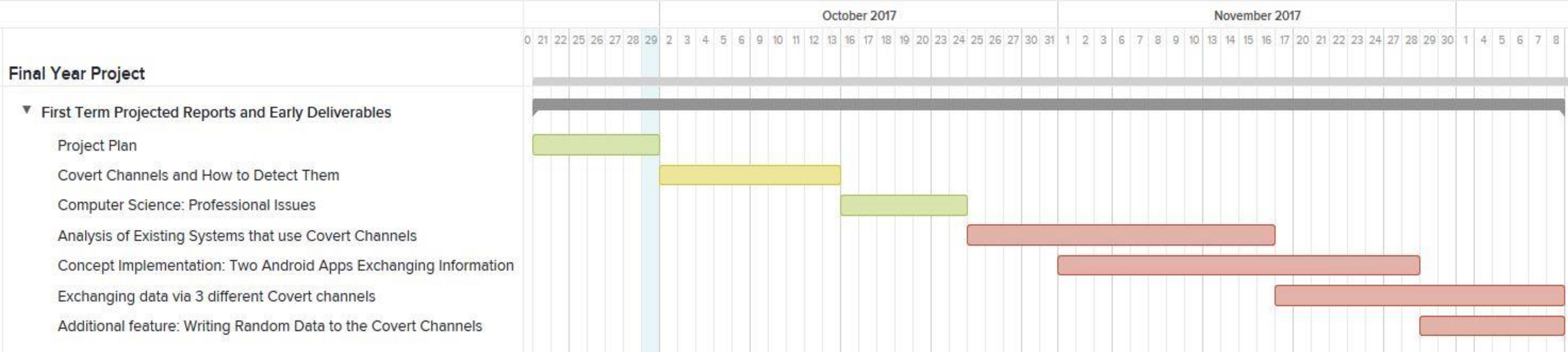
My project is, in short, about creating an app that can identify which other apps on that device are using covert channels to communicate. Including the covert channels mentioned above and additional channels such as battery percentage and phone call logs. This project is only being implemented on the android operating system due to the design of the android OS and it’s mobile interpretation of UNIX. I also plan this app to be backwards compatible to android 5.0 (Lollipop). This may easily be achieved as the software I’m using (android studio) has many emulators that allow me to test my program on devices across many different versions of the OS.

Originally, I wasn’t aware that a floor such as covert channels existed in a widely used and popular OS like android, however after researching more into the subject, I found it fascinating the flaws that covert channels give day to day smartphones and the insecurities it gives personal data that we all have on our smartphones. Sensitive data such as doctors’ appointments and details related to the user can be sent from a private network disabled app to one that has access to the internet, posing a huge risk for the user’s data.

Over the entire year, my goals are to create an app that detects possible covert channels being used, as well as two apps that actively communicate with each other using covert channels. These two apps will test whether the detection works, and while the UI on the main app will clearly and cleanly show the user what is at risk. Another goal is to avoid other apps using the covert channels but sending random data through them, however this isn’t essential to the project. When making these apps, I must be weary of possible flaws that may occur: if the device has not great hardware, the application that I make may use most of the CPU memory, making the device nearly unusable and hence making the applications pointless to a system.

This project is exceptionally interesting to me as I plan to continue android, and other mobile Operating systems, security in the future and possibly advance a professional career in that area. Thus, the reason I chose ‘Detecting Android Covert Channels’ as my project.

Timeline



Final Year Project

▼ First Term Projected Reports and Early Deliverables

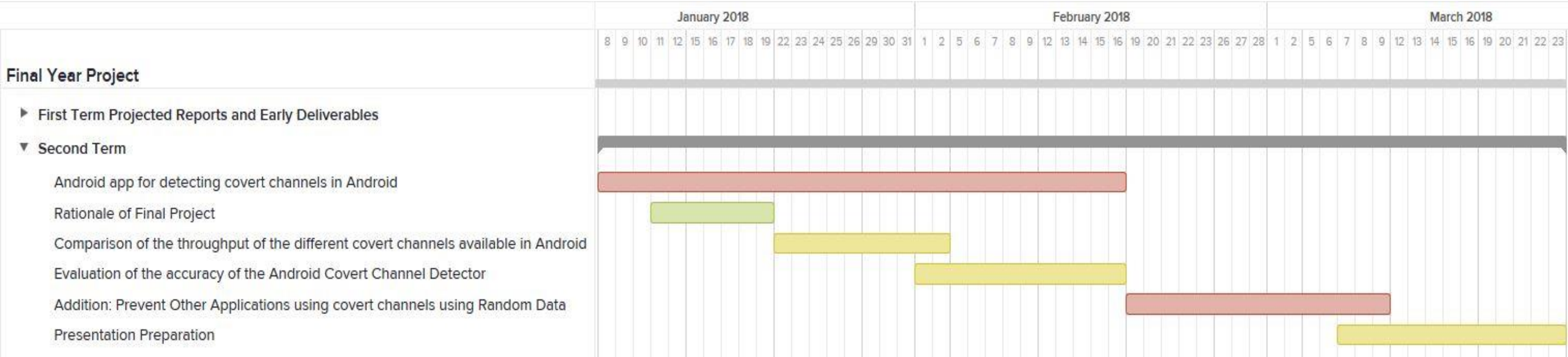
- Project Plan
- Covert Channels and How to Detect Them
- Computer Science: Professional Issues
- Analysis of Existing Systems that use Covert Channels
- Concept Implementation: Two Android Apps Exchanging Information
- Exchanging data via 3 different Covert channels
- Additional feature: Writing Random Data to the Covert Channels

Explanations on Reports and Proofs of Concept

As you can see in the Gantt Chart above, during the first term, I plan to complete three reports and three proofs of concept (not including the project plan). The start and end dates are on the chart with the colours representing the predicted difficulty: red indicating difficult, amber indicating probably normal difficulty and green possibly easy.

- In further depth than before, my first report is: Covert Channels and how to detect them; which I have given myself two weeks to write due to the depth I plan to go into (02/10 - 13/10). This report will go into the current theory and knowledge of Covert channels, analysis and review of inter process communication with coding extracts of how to send and receive through said channels. Also included is the covert channels detection theory's and which may I apply in my project. As you can see, I've coloured it amber to indicate the predicted difficulty of the report.
- The next report is: Computer Science-Professional Issues. I've only given myself a week and a half (and coloured it green) as I predict it's going to be a simple report to write (hence 16/10 – 24/10). This will be discussing the possible issues that computer scientist face, this can include licencing, cost of piracy and even plagiarism. This can even include ethics of writing high risk programs and the ethics of handling private and confidential data correctly.
- The report starting on the 25/10 and ending on 16/11 is: Analysis of Existing systems that use Android Covert channels. This report that I added in, will be discussing and analysing existing systems. Discussing their efficiency, whether they could have coded it a different way that would have been easier and what covert channel's do they use and could they have used. This will hopefully help in deciding what and how I should code my own software and decipher what covert channel's I will use.
- This next task of this term is one of the proofs of concept, creating two android apps that can send and receive data through covert channels (1/11 – 28/11). This will be conducted with vertical slicing with a top down approach. I will start by designing for the latest android version and work backwards, testing down to android 5.0 (using TDD on the way). I will code using a volume channel, as it seems to be the most researched.
- This next task that I planned to create is adapting the previous task to send and receive data through three different covert channels whether that is a storage channel or a timing channel (17/11 – 8/12). This is an extra I added on that may prove to be difficult however I believe down to the reports I will have done in the previous weeks, I should be able to implement this feature. This may include extra UI's to differentiate between the different covert channels used.

- The last task I planned to do is, again, an additional feature. Writing random data to the covert channels (29/11 – 8/12). This is to aid the second term task of writing random data to covert channels in order to avoid other apps using them.



Final Year Project

▸ First Term Projected Reports and Early Deliverables

▾ Second Term

Android app for detecting covert channels in Android

Rationale of Final Project

Comparison of the throughput of the different covert channels available in Android

Evaluation of the accuracy of the Android Covert Channel Detector

Addition: Prevent Other Applications using covert channels using Random Data

Presentation Preparation

- The first task in second term will be the overlying project: creating an android app that detect covert channels. Similar to before, this will be using TDD, top down approach of a vertical slice. Hence the first task in this application will be designing the UI accompanied with some background code. The start date is 8/01 and finished 6 weeks later, on 16/02.
- Rationale of Final Project will be written between 11/01 and 19/01 due to the ease of this task. It will contain the aims, objectives and the structure of the final report.
- Comparison of the throughput of different covert channels available in android is where I will review and analyse the current bps throughput and compare the result across multiple different channels. Once taking in multiple factors, and concluding which I believe to be the best to communicate through, the analysis will be documented in a report (dates from 22/01 – 2/02).
- The next set of analysis that I'll do is: Evaluation of the accuracy of the android Covert Channel Detector. This will be critiquing my own project on: how well it runs, how much space does it take up on a device, does it use a reasonable amount of resources, and of course does it detect android covert channel's use (in one or many covert channels). This will be executed between 1/02 and 16/02
- This next task will be adapting my detector into actually avoiding other applications using the covert channels by sending random data down the channel. This, with the help of the first term sending random data bits through the channel task, will be given three weeks due to the complexity (19/02 – 9/03). This will have a separate UI form and hence will be a separate vertical slice in the coding process.
- The last task will be the presentation preparation which will contain working on the presentation and demo. This will be from 7/03 up until the deadline 23/03. This will a process of making a presentation and speech to accompany it as well as many practises of the demo and how the application will be shown.

Risk Assessment

Specific Project Risks

- One of the possible risks that my project could face is that the three applications may use too much of the RAM on the emulated device. When running the device on my computer, it won't fail but applying it to an actual device with, for example, <500Mb of RAM, the device may falter in the processing. This is an issue that may occur (although unlikely) and would be difficult to resolve (halting the timeline), and therefore needs to be thought about when designing the applications. If this does occur, then must be considered high importance.
- Android application not scaling to different versions of android. As I mentioned before, I plan to make each application backwards compatible to android 5.0, however due to time issues or the inability to scale the application, this minor aim may not work and hence is a risk of the project. It has a medium likelihood however is not important enough to halt the coding process.
- Issues with permissions to an individual device. As mentioned in (Wade Gasior n.d.), a user or security software may be alarmed if an application asks for too many permissions. Therefore, I need to be weary of this issue when it comes to programming the two applications (sender and receiver). This would be considered relatively likely and would majorly halt the coding process so must be considered high importance. This will need to be thought about when designing the software, to prevent this from happening.
- Due to the java language not allowing access and manipulation of raw sockets, for network covert channels, it is limited to high level sockets only, which may make it more difficult to obtain access/code. This is of medium to high importance as it'd mean I'd need to change the way I was coding the sockets.
- For this project, I am using android studio which is new software I haven't used for major projects before. A risk could be not knowing all the in-built features and therefore getting used to the software may halt the coding process. I'd consider this low importance due to the number of online tutorials you may get with this software, even though it'd be a probable situation.
- Areas with low bandwidth could be a risk due to the detector not being able to detect network covert channels, which would make it difficult to analyse and prevent. This is High importance but low probability, due to the resources the university provides high speed internet.

General Project Risks

- Delays in work. Project delays are a common risk as things don't always go to plan. I'd personally say this is fairly likely and relatively important however my timeline incorporates the possibility of being delayed.
- Data loss when not using svn or githib is a major issue when working on a project. Not only can you access your project from anywhere, your physical machine may damage, therefore destroying your only copy. This is unlikely risk however an incredibly important risk.
- The testing device may break. I'll be using emulators and my own tablet, and if my tablet breaks, I'll have no real-world device example of my program working (especially when testing the program doesn't use too much RAM. This is of high importance but low probability.

Bibliography and Analysis

Ahmed Al-Haiqi, Mahamod Ismail, and Rosdiadee Nordin. 2014. "A New Sensors-Based Covert Channel on Android." *www.hindawi.com.* September 14. Accessed July 2017. http://dx.doi.org/10.1155/2014/969628.

Roman Schlegel, Kehuan Zhang, Xiaoyong Zhou, Mehool Intwala, Apu Kapadia, XiaoFeng Wang. 2011. "Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones." *NDSS Vol 11* 17.

Swarup Chandra, Zhiqiang Lin, Ashish Kundu, and Latifur Khan. 2014. "Towards a Systematic Study of the Covert Channel Attacks in Samrtphones." *Internationsal Conference on Security and Privacy in Communication Systems.* Springer International Publishing. 22.

Wade Gasior, Li Yang. n.d. "Exploring Covert Channel in Android Platform." 5.

Analysis of readings

At first, I read S. Chandra's "Towards a Systematic study of the Covert Channel Attacks on Smartphones", and one of the most notable pieces of information was a table of what parts of a system had the ability to be used as a covert channel and had been researched, and which had not been researched. This article, as well as informing me of concepts I had not known about, helped me to identify what covert channels I would want to use in the future for my early deliverables. While also excluding some that had crossed my mind, but had not been researched enough for me to warrant doing a project on it.

Ahmed Al-Haiqi's "A new sensor based Covert Channel on Android" developed my knowledge to the project as it directed me towards the shared Resource Matrix and the non-interference method to uncover covert channels (methods that I hadn't seen been mentioned in S. Chandra's article). While giving a list of covert channels and possible countermeasures, which inclined me to want to make my report 'covert channels and how to detect them' first as it would deepen my knowledge before I'd have to code the applications. I highly value this article as I will use the table of covert channels with throughput (bps) and possible countermeasures throughout both terms for my project.

Wade Gasior and Li Yang's "Exploring Covert channel in Android Platform" solidified my knowledge on the implementations of the 2 main types of channels. The article split a model based timing channel into 4 phases, page two (Wade Gasior n.d.). After explaining in depth, the two main Covert channel types, and that was notable about this article, was it explained the implementation challenges (page 3) in reasonable depth. I proceeded to give myself more than two weeks for my implementation due to this factor. For example, one of the challenges was, when installing the applications, permissions must be granted by the user, which may raise user suspicion as well as security software identifying these application as 'over-privileged' thus raising the security software's suspicion. For my software, I will have to evade these suspicions without asking too many permissions from the user.

Roman Schlegel's "Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones." Is an article that is referanced by all other arcticles and journals that I've read. I particulaly found this article note worthy as it describes the soundcomber app that was created and how it works, including useful diagrams about the collection and transmission of data. Also with a section about the Defence actitecture, page 13, which impacted my timeline as I feel I need as much time as possible to implement the detectetion application. This article has impecable value to my project due to the sheer amount of data it has provided about accuracy of different sensors that I may need to manipulate.