Sri Lanka Institute of Information Technology

Evaluation of Trust in Android Devices

Project ID: **16J-008**

Project Proposal
Comprehensive Design & Analysis
June Intake 2016

B.Sc. Special (Honors) Degree in Information Technology

Submitted on Wednesday, 3rd of August, 2016

Authors:

Student ID	Name	Signature
IT13023256	Jayasekara R.A.N.T.	
IT13067816	Priyadarshani T.D.H.	
IT13048624	Chathuranga K.B.L.	
IT13096908	Anooj R.	

Supervisor

…….
Mr. Lakmal Rupasinghe

Co - Supervisor

…….
Mr. Krishnadeva Kesawan

Declaration

"We declare that the project would involve material prepared by the Group members and that it would not fully or partially incorporate any material prepared by other persons for a fee or free of charge or that it would include material previously submitted by a candidate for a Degree or Diploma in any other University or Institute of Higher Learning and that, to the best of our knowledge and belief, it would not incorporate any material previously published or written by another person in relation to another project except where the acknowledgement is made in the text."

……	……
Jayasekara R.A.N.T.	Priyadarshani T.D.H.
……	……
Chathuranga K.B.L.	Anooj R.

ABSTRACT

The Internet of Things (IoT) is one of the new emerging technologies. The main limitation that might threaten the growth of IoT will be the difficulties in the Security. Since IoT is composed of various heterogeneous smart devices, the security vulnerabilities in IoT are very high because the number of malware that can affect will also be higher. Although there are many studies carried out on trust evaluation, normal trust evaluation methods will not serve due to the high diversity of the IoT devices. This paper state of a machine learning mechanism to determine the trust of the devices involved in IoT by looking at various aspects that indicate the presence of malware in a device which will ultimately diminish trust. The features we collect are based on Network Information, Application information, Operating System Information and User Information. By considering many parameters from these features, the overall trust of the device is determined. Our ultimate goal is to build a trust network by connecting devices with higher trust values, so that sensitive information can be exchanged through IoT. These computations will be done using a Machine Learning algorithm and we plan to present simulation results that support our work.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

Internet today, is not what it was yesterday or it was a decade back. And it will not be the same tomorrow or a decade ahead. It is a fast growing technology that has no bounds. Currently the world is focused on Internet of Things (IoT) where 'things' could be devices of all types and kinds, ranging from consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects [1], i.e. IoT is a network of these physical objects which allows them to be connected and interacted with other objects in the network, with minimized human interaction. And thus, a large number of sensitive data are exchanged through devices on a daily basis. Since our work is focused on Android Mobile devices, we will be talking about them from this point forward.

The usage of Android Mobile devices are still on the rise, so are the facilities they provide. From sending a simple text message to channeling doctors and online banking transactions to all sorts of services imaginable. In 2015 Q2 (2nd Quarter) Android devices dominated the smartphone market with a share of 82.8%, and that's a 13.5% increase when compared to 2012 Q2 [2]. Having the largest market share also makes Android, the most attractive platform for malware. According to G DATA statistics 440,267 new Android Malware were identified in 2015 Q1 and that's a significant 21% increase compared to 2014 Q1 [3].

According to the survey research done by [4], the trust of a particular node is a subjective assessment by an agent/other peer node on the reliability and accuracy of information received from or traversing through that node in a given context. Trust reflects the belief or confidence or expectations on the honesty, integrity, ability, availability and quality of service of target node's future activity/behavior. It also reflects the mutual relationships where a given node behaves in a trustworthy manner and maintains reliable communications only with nodes which are highly trusted by the given node. In our case, nodes are considered as devices. We plan to clarify these concepts in a way that it will benefit our project.

1.2 Literature Review

Numerous studies were carried out to determine the device trust of Android devices considering various aspects. Most of them are concerned on MANETS and peer-to-peer networks. And there are limited numbers of studies carried out on Device trust for IoT. Some of the studies are determine device trust based on Malware Analysis and some of them are based on other factors such as User's information.

A study carried out by Weiss and Reznik [5] on Trust Evaluation in Mobile Devices stated that Trust evaluation should integrate various metrics ranging from accuracy and reliability of the data sources to the security of the procedures and tools used. In order to fill these criteria they used application details, device feature security, sensor security, Battery Usage, CPU Usage, Network usage and level of privacy provided by the device as parameters. All these parameters are considered for the identification of malicious behavior. Since malicious program tend to use up more resources this method is effective. Yet they have failed to capture the user's details which represent trust just as much as other factors.

Zhao and Pan [2] introduced a Machine Learning based Trust Evaluation Framework for Online Social Networks. They have considered numerous factors about users such as the density of interactions between the two users, the similarities between the two users, the number of mutual friends between the two users and etc. Although this is a good approach they only give the final output as two values, i.e. whether we could trust or distrust a particular device. This approach isn't ideal to be used in an IoT environment or MANETS because we cannot completely disregard all the untrusted devices. It will degrade the performance. So rather than completely saying a particular user or device is untrusted, assigning a continuous value as the final output is more applicable cause then it allows us to compare.

Bao and Chen [4] introduced a Dynamic Trust Management for the IoT Applications. They provided a flexible and accurate trust assessment for IoT entities. Although their trust

assessment is quite accurate and successful they have failed to detect the presence of malicious behavior and selfish nodes. They have mainly focused on user's relationships.

Trust or Reputation analysis is looked upon in various angles. Presence of malicious programs affects trust of the device most because most of the time the user is unaware about the presence of malware. We believe that presence of malware can be analyzed and confirmed in various aspects. Various studies are based on malware detection and analysis. There are static analysis techniques, dynamic analysis techniques as well as hybrid approaches. There are multiple ways how malware get installed into devices and several different ways they get activated. They could exploit the vulnerabilities in the user level, network level, operating system and applications installed in the device. Therefore it is necessary to identify the weakness at each of these levels because all of these weaknesses will diminish the trust of the devices and provide many opportunities for malware to get installed in our devices.

So far there are large numbers of Android Malwares roaming around. ''Among all mobile malware, the share of Android based malware is higher than 46% and still growing rapidly." [6] These malware are diverse and how they get activated and the consequences they cause and where they cause these consequences are varied. Therefore in order to identify all of these malware present in a device, looking at only one aspect won't work.

Weaknesses Malware uses:

- Using System events and System calls
- Vulnerabilities in the OS such as bugs, configuration oversight, designation
- Vulnerabilities in the applications installed such as bugs, configuration oversight, and designation
- Some Malicious Payloads cause Remote Control. This can be analyzed using network information [6].
- Some Malicious Payloads cause Financial Charges, by subscribing the user to premium rate services without user's awareness. These things happen in the

background. Therefore it is necessary to detect how much data or resources a particular app takes up in the background.

Yerima and Sezer [7] introduced a Malware Detection Mechanism using Parallel Machine Learning classifiers. As the features for the feature vector they extracted API related features, App permissions and Standard OS and Android framework commands. Although they have considered both application and OS related information to detect the malware they have neglected the network and user related information which is just as important.

Mekouar, Iraqi and Boutaba[8] brought forward a method called Detecting Malicious Peers in a Reputation-Based Peer-to-Peer System. They allocate the reputation based on the feedback and Authenticity of the files given. Although this is a good approach to be used in a peer-to-peer system we cannot use this in an IoT environment because it's not practical to collect feedback.

	Product/Paper				
Feature	Andromaly	[9]	[8]	Trust Evaluation in mobile devices	A Machine Learning based Trust evaluation
Ideal for IoT environment.	N	N	Y	N	N
Parameters for the computations can be easily obtained.	Y	N	Y	N	Y
User level parameters considered.	N	N	Y	N	Y
Presence of malware is considered.	Y	N	N	Y	N
Selfish nodes are detected.	N	Y	N	N	N
Malicious nodes are detected.	N	Y	N	N	N
Network parameters (Packet Size, No. of Packets, Duration of a Transaction, No of Transactions Completed) are considered.	N	N	N	Y	N
Vulnerabilities in Android OS is considered.	Y	N	N	Y	N

Vulnerabilities in the apps installed are considered	**Y**	**N**	**N**	**Y**	**N**
Overall trust value for the device is given	**N**	**Y**	**Y**	**N**	**N**
Creates a secure network	**N**	**Y**	**N**	**N**	**Y**

1.3 Research Gap

Most products and papers are, concerned only about one or few parameters that affect the trust.

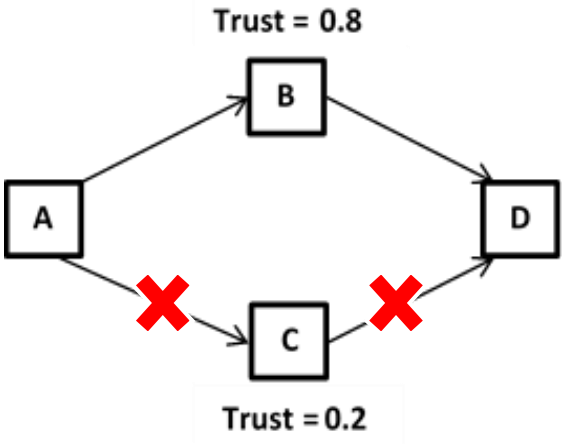
After the overall comparison of all products and papers, we identified that the products have used only a few parameters/factors to evaluate the trust value. For example In IoT Security research paper, they only have mentioned about selfish nodes and misbehavior nodes and its specification. In another research paper named Trust of Android devices, they only have mentioned and evaluated trust value through hardware, software and privacy level of the android devices. In another research article about Malware analysis, they only have mentioned malware characteristics and its specification. Additional to these parameters/factors, there are some other related parameters that are needed to evaluate trust value of an Android device. Our team has identified all parameters and categorized those parameters into four major categories. Those are trustworthiness of the user, trustworthiness of the network, vulnerabilities in the Android OS, vulnerabilities in the applications installed.

- Lack of an ideal system to be used in the IoT environment conditions

  Due to universal interconnectivity in the IoT environment, the devices connected should have a reliable, accurate and efficient communication link among each other. If a device has various security vulnerabilities, it does not only affect the device itself, but also affects the users, networked devices and the whole IoT environment's infrastructure.
- Not continuous value generated for trust

  In most of the existing systems related to our topic, they had only identified, whether a particular device/node in the network, "can be trusted" or "cannot be trusted". So when determining the best path for the communication, due to the above scenario, if all the paths have untrusted nodes, the communication will not take place. But having a continuous value for trust will help determining the best path have a threshold value.

For example:



If A, wants to communicate with D, A can reach D through B as well as C. but if A is sending sensitive information to D, it should reach D through B because B has the highest trust value.

1.4 Research Problem

- IoT consists of numerous heterogeneous devices
- Security is the main threat for the growth in IoT
- Evaluating the trust for devices in the domain helps to make a secure IoT environment.
- Thus, sensitive information can be exchanged

2. OBJECTIVES

2.1 Main Objectives

We have identified our main objectives to be achieved.

- To be able to analyze malware present in devices,
- Using the malware analysis to determine the overall trust of the device.

2.2 Specific Objectives

- To extract user level information
- To extract network information
- To identify vulnerabilities in the Android OS
- To identify vulnerabilities in the applications installed.

3. RESEARCH METHODOLOGY

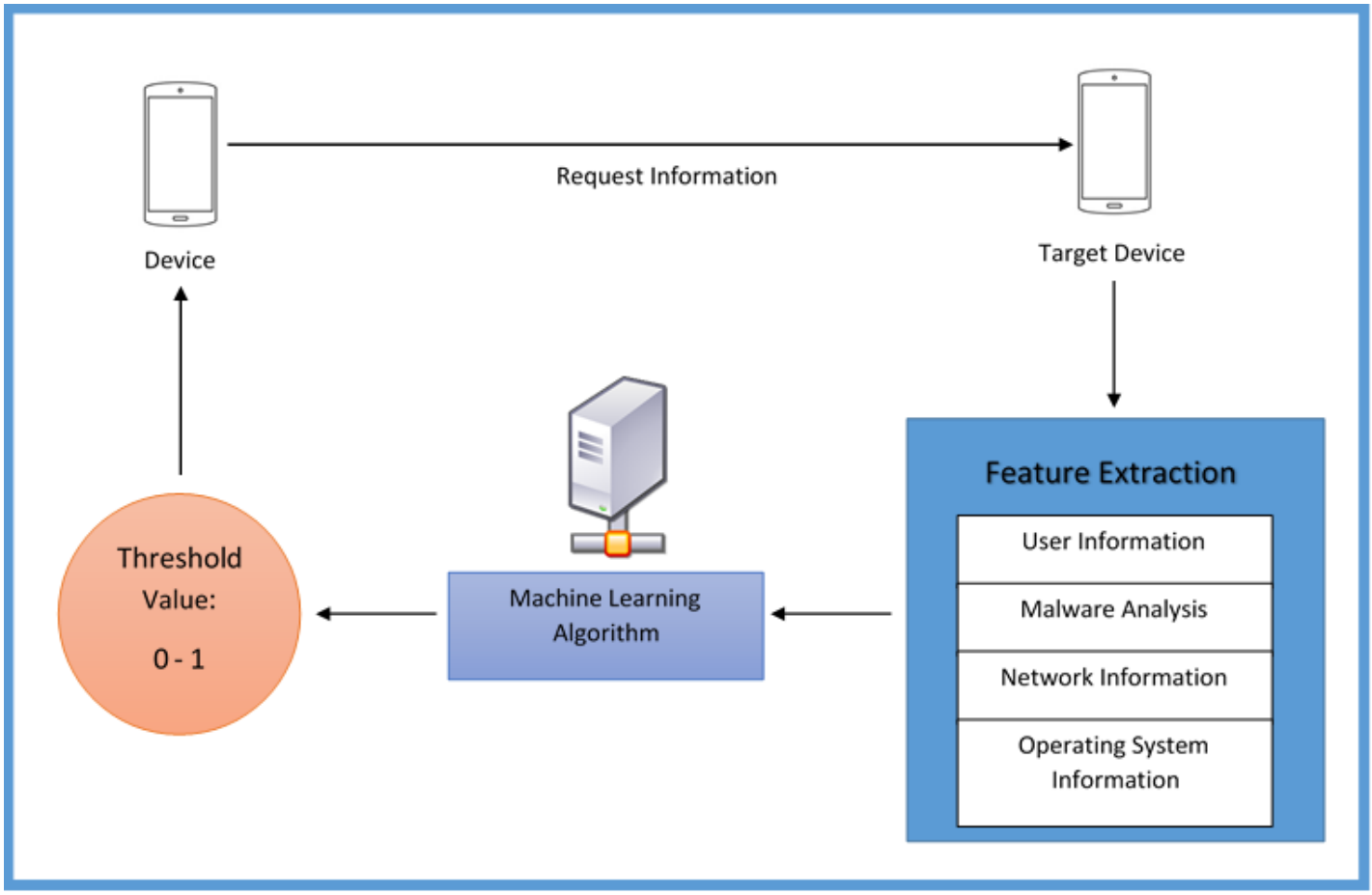
3.1 Methodology of the Proposed System

In this section, we will discuss, how we plan to do the project, throughout the software development life cycle. An overview of the proposed system will be on section 3.2, and in section 3.3, for each step in software development life cycle, we will discuss the process. In section 3.3.4, we will discuss, how each member of our team, will individually contribute to the project.

Our proposed system has 4 main sub – components.

I. Compute the trustworthiness of the User, in the overlay network
II. Compute the trustworthiness of the Nodes, in the network
III. Compute the trustworthiness of the Android OS
IV. Compute the trustworthiness of the Applications installed in the device.

3.2 System Architecture



3.3 Development

3.3.1 Compute the trustworthiness of the User, in the overlay network

Most smart objects in IoT are human-carried or human-related smart devices [4]. Therefore the kind of owners of these smart devices also impact for the security. Even If the device is secured using various mechanisms and there's no malware in the device we cannot guarantee whether the sensitive information we pass through these devices is safe unless we are confident the owner of the device is trustworthy. Although the device may seems benign, user maybe malicious. It is pointless to asses trust based on the technical aspect of the device if the owner of the device is not trustworthy. Hence trust of the owner of the devices should be considered when determining the overall device trust.

In order to determine what sort of user a particular device belongs to we have to obtain essential information such as:

- Whether two users are friends–since friends are more likely to be trusted than strangers.
- The designation of the user.
- Total number of friends a user has- If the user is trusted he/she is assumed to have more friends.
- The similarities between the two users – users tend to trust other users with more similarities
- The Frequency of interactions of the two users- If two uses interact frequently it represent the intimacy of their relationship
- The distance between the users- If two users live close by there's a higher chance they might know each other and trust each other.

Since almost all of the people now use social media this information is easily available. Many Studies have been carried out to calculate the device trust using sociological information, and to some extent they have been quite successful. Most of the studies are carried out for MANETS or peer to peer networks [2, 3]. And there is very limited number of studies based on sociological features for IoT [4].

Zhao and Pan [2], introduced a machine learning based trust evaluation approach for online social networks using structural and behavioral information which can be easily retrieved from social network data, having only two states for the trust level -Trust or Distrust. The features they considered for the feature vector are:

- Number of followers
- Number of friends
- Ratio of Friends to followers
- Number of bi-directional followers
- Ratio of Number of bi-directional followers to total number of followers.
- Distance from one node to the other
- Similarity of the two nodes
- The interaction of two nodes

If the feature vector falls between 0-0.5 it is called a distrust device and if the feature vector value falls between 0.5-1 it's known as a trusted device.

Although these conditions are ideal for a trust evaluation of a social network, same conditions cannot be used for IoT cause

- In IoT out of many devices available, we have to know which device we could trust most using the highest overall trust value. Since this method simply state whether to trust or distrust a device, which is not very helpful.
- For this study we are planning to use Facebook as the social media to obtain the owner's information. Some of the features that contribute to the feature vector in the above study are not available on Facebook.

Welikala [3] proposed a trust based mechanism to pass confidential information through a social network ideally sensitive health related information using Bayesian Belief Networks.

Bao and Chen [4], introduced a dynamic Trust management for IoT applications by considering the relationships among device owners. This protocol is made up of Social

Trust + Direct Observations+ Indirect Recommendations. For direct observations honesty, co-cooperativeness and common interest were considered.

By going through above studies which is developed to determine trust based on social media information, the required information that needs to be extracted through social media is decided. They are listed below:

If there are two users called x and y:

- Whether x is a friend of y
- The designation of x
- The similarities between the two users
- The density of interactions of x and y
- The distance between x and y
- Ratio of Mutual friends to Total number of friends

All these parameters are the parameters of the other contributing factors make the feature vector. This feature vector is passed through a machine learning algorithm to determine the overall trust based on the degree of malware present in the device.

Social Network information data can be extracted using specialized software which yields a Graph Description Format (GDF) file as the result. Although there were tools like Netvizz which is ideal for Facebook data extraction it has been disrupted by Facebook due to privacy issues. API endpoints of these social platforms could be used and process the result with a language like python but it is a tedious task [5]. Due to these difficulties we are currently searching for better open source data extraction software.

The output of the data extraction software can be analyzed using Social Network Analysis techniques [3]. Social Network analysis is used for formulation and solution of problems that have a network structure, such structure is usually captured in a graph. Using graph theory, analytical tools and software developed ideally for the visualization and analysis of social media [6]. There are specialized tools built for this purpose such as Gephi, Centrifuge and EgoNet etc. For the development of this project Gephi will be used since it

is open source and it allows understanding and exploring of graphs to reveal hidden properties [7].

There are certain metrics used in social Network analysis which is helpful to derive the required parameters for the feature vector. These metrics are:

1. Connections
    - Homophily – The degree the nodes bonds with similar nodes and dissimilar nodes
    - Multiplexity – Relationship Strength/ the degree of similarities between the 2 users.
    - Mutuality/Reciprocity – The density of the interactions
    - Network Closure/Transitivity – Their friends are also friends
    - Propinquity – The chance of nodes being close to other nodes that are geographically close
2. Distributions
    - Tie Strength – Intensity of the bond between two nodes.

3.3.2 Compute the trustworthiness of the Nodes, in the network

- Hop count between the two devices
- Number of packets transferred
- Number of packets received
- Size of a packet
- Number of completed transactions
- Duration of a transaction
- Presence of Selfish nodes

Currently we are planning to use the IP header and the Watchdog mechanism to extract these information

3.3.3 Compute the trustworthiness of the Android OS

First, the operating system is checked to confirm that it is running the most recent version available. If devices may not receive current android and security updates from Google and the devices without the necessary updates may become vulnerable to threats. An algorithm looks for current android version and security updates on Google, compares with installed android version and security updates in the device. After the verifications of all related information, it generates a trust value score which lies on android version. (For example if android version and security updates in the device are up to date, it will get good score.) E.g. Score 01

Second, the personal security settings on the device are examined to determine if the user is utilizing the appropriate tools to secure the device. These security software take different approaches in their design and implementation, which lead to different detection ratio even for same security problems. An algorithm will collect related information of security tools and user settings on android device, it generates a trust value score which lies on user settings on security tools. (For example if the device uses LookOut security tool and user enables every security option in the tool, it will get good score) E.g. Score 02

3.3.4 Compute the trustworthiness of the Applications installed in the device

The first step of an algorithm lists all of the applications installed on the device which are directly downloaded from Google Play store. After that the manifest file for each application installed is analyzed in order to fetch the application name, package name, required features, version, required permission, path info, date on which the application was installed and the target SDK version.

The second step of algorithm generates a trust value score for installed application. Google Play store holds all data associated with the distribution of an application including its APK file and associated documentation. Following information can be retrieved from Google Play store:

- Number of installs: Total number of installs across the application life

- Number of reviews: Total number of reviews from unique users
- Score: User rating of 1.0 to 5.0
- Developer: Name of the developer
- Permissions: Which resources can be accessed by the application?

The first three fields can be used to find an applications popularity that, when matched with a history of values, shows user trend information. Above information will be retrieved by the algorithm and generates a trust value score which lies on application from Google Play store. (For example if the application was from an unknown publisher with low score and low number of downloads, it will get low score.) E.g. Score 03

The third step of algorithm, the installed application from Google Play store is checked to confirm that it is running the most recent version available. If devices may not receive application updates from Google Play store and the devices without the necessary updates may become vulnerable to threats. Hence the algorithm looks for current application updates on Google Play store, compares with installed application version in the device. After the verifications of all related information, it generates a trust value score which lies on application version. (For example if application version in the device is up to date, it will get good score.) E.g. Score 04

The next step of algorithm lists all of the applications installed on the device which are downloaded from other application markets. After that the manifest file for each application installed is analyzed in order to fetch the application name, package name, required features, version, required permission, path info, date on which the application was installed and the target SDK version. After the verifications of all related information, it generates a trust value score which lies on applications from other application markets. (For example if the application couldn't retrieve above information, it will get low score.) E.g. Score 05

The last step of algorithm, the installed application's data usage level verification. The nature of mobile data connectivity is reflected by various indicators of data activity, thereby allowing any app to detect when other apps transfer data over the mobile interface. Android applications uses mobile data for two different process:

I. Foreground process

II. Background process

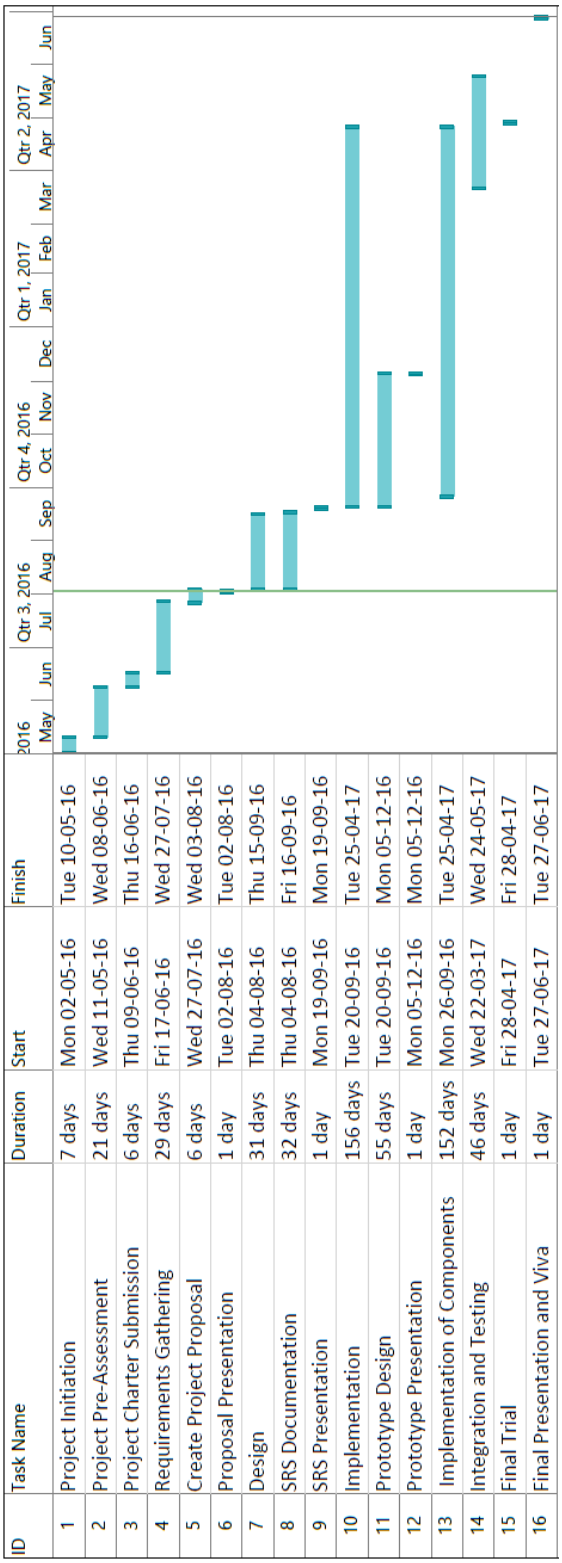
Although an installed application is trusted, verification of data usage level also will confirm that the application's trustworthiness. To make sure the verification, algorithm does compare foreground data usage level and background data usage level and generates a trust value score which lies on data usage level by applications. (For example if an application uses more mobile data for foreground process than background process, it will get good score) E.g. Score 06

- All these features are extracted and a feature vector is built.
- The finalized feature vector is analyzed using a machine learning technique
- The machine learning technique we plan to use is "Deep Reinforcement Learning"

  Reason:

  - Helps to determine the ideal behavior within a specific context, in order to maximize performance
  - It uses trial and error learning

3.4 Gantt Chart



ID	Task Name	Duration	Start	Finish
1	Project Initiation	7 days	Mon 02-05-16	Tue 10-05-16
2	Project Pre-Assessment	21 days	Wed 11-05-16	Wed 08-06-16
3	Project Charter Submission	6 days	Thu 09-06-16	Thu 16-06-16
4	Requirements Gathering	29 days	Fri 17-06-16	Wed 27-07-16
5	Create Project Proposal	6 days	Wed 27-07-16	Wed 03-08-16
6	Proposal Presentation	1 day	Tue 02-08-16	Tue 02-08-16
7	Design	31 days	Thu 04-08-16	Thu 15-09-16
8	SRS Documentation	32 days	Thu 04-08-16	Fri 16-09-16
9	SRS Presentation	1 day	Mon 19-09-16	Mon 19-09-16
10	Implementation	156 days	Tue 20-09-16	Tue 25-04-17
11	Prototype Design	55 days	Tue 20-09-16	Mon 05-12-16
12	Prototype Presentation	1 day	Mon 05-12-16	Mon 05-12-16
13	Implementation of Components	152 days	Mon 26-09-16	Tue 25-04-17
14	Integration and Testing	46 days	Wed 22-03-17	Wed 24-05-17
15	Final Trial	1 day	Fri 28-04-17	Fri 28-04-17
16	Final Presentation and Viva	1 day	Tue 27-06-17	Tue 27-06-17

4. DESCRIPTION OF PERSONAL AND FACILITIES

Following table shows of personal facilities as well as shows their responsibilities. The responsibilities assigned to each group member according to their skills, interest and the capability of doing that task

Member	Component	Task
IT13023256 Jayasekara R.A.N.T	Extract Network Information	• Hop count between the two devices • Number of packets transferred • Number of packets received • Size of a packet • Number of completed transactions • Duration of a transaction • Presence of Selfish nodes
IT13067816 Priyadarshani T.D.H.	Extract User Information	If there are two users called x and y: • Whether x is a friend of y • The designation of x • The similarities between the two users • The density of interactions of x and y • The distance between x and y • Ratio of Mutual friends to Total number of friends
IT13048624 Chathuranga K.B.L.	Identify vulnerabilities in the installed applications	• Number of times a particular application has been installed • Number of reviews for a particular application • Is the application up-to date • User rating a particular application has received • Is the Developer recognized
IT13096908 Anooj R.	Identify the vulnerabilities in the Android OS	• Is the device updated • Appropriate security tools are utilized • Mobile Data usage in the background processes • Mobile Data usage in the foreground processes • Number of System Calls made by a particular app

5. BUDGET

	Rs.
A4 Sheets	500.00
Photocopies	2000.00
Printouts & Bindings	3000.00
Internet Charges	5000.00
Telephone Charges	3000.00
Total Cost	13500.00

6. REFERENCES

[1] K. Rose, "Internet Society Releases Internet of Things (IoT) Overview Whitepaper: Understanding the Issues and Challenges of a More Connected World | Internet Society", Internetsociety.org, 2015. [Online]. Available: https://www.internetsociety.org/blog/public-policy/2015/10/internet-society-releases-internet-things-iot-overview-whitepaper. [Accessed: 30- Jul- 2016].

[2] "IDC: Smartphone OS Market Share", www.idc.com, 2016. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed: 31- Jul- 2016].

[3]"G DATA Releases Mobile Malware Report for the First Quarter of 2015", G DATA Software AG, 2016. [Online]. Available: https://www.gdata-software.com/g-data/newsroom/news/article/g-data-releases-mobile-malware-report-for-the-first-quarter-of-2015. [Accessed: 31- Jul- 2016].

[4] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 279-298, 2012.

[5] 2016. [Online]. Available: https://www.quora.com/Which-tools-are-available-to-collect-social-network-data. [Accessed: 03- Aug- 2016].

[6]"Social Network Analysis", Slideshare.net, 2016. [Online]. Available: http://www.slideshare.net/gcheliotis/social-network-analysis-3273045. [Accessed: 03- Aug- 2016].

[7] D. Desale, "Top 30 Social Network Analysis and Visualization Tools", Kdnuggets.com, 2016. [Online]. Available: http://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html. [Accessed: 03- Aug- 2016].

[8] I. Chen, F. Bao, M. Chang and J. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 5, pp. 1200-1210, 2014.

[9] L. Mekouar, Y. Iraqi and R. Boutaba, "Peer-to-peer's most wanted: Malicious peers", Computer Networks, vol. 50, no. 4, pp. 545-562, 2006.

7. APPENDICES

- IoT – Internet of Things
- Q1 – 1st Quarter of the Year
- Q2 – 2nd Quarter of the Year
- MANET – Mobile Adhoc Network