International Journal of Engineering Sciences & Emerging Technologies, Dec. 2012.
ISSN: 2231 – 6604 Volume 4, Issue 1, pp: 42-50 ©IJESET

AN INTEGRATED REQUIREMENT-BASED FRAMEWORK TO EMPHASIZE SECURITY SYSTEM: ARGUE WITH ANDROID SECURITY FOR VALIDATION

Md. Faisal[1], M. Mohideen Abdul Kader Jailani[2], Sanjai Gupta[3], Mohd. Hussain[4]

[1]Department of Information Technology, Nizwa College of Technology, Nizwa, Oman
[1]faisal31621@gmail.com
[2]Department of IT, Nizwa College of Technology, Nizwa, Oman
[2]jailani.bnu@gmail.com
[3]Department of Computer Science, CMJ University, Shillong, India
[3]guptasanjay3@gmail.com
[4]Department of Computer Science & Engineering, M G Institute of Management & Technology, Lucknow, UP, India
[4]mohd.husain90@gmail.com

ABSTRACT

The goal and objective of the purposed framework is to bring requirement engineering and security requirement engineering process together in such a simplify form so, that a requirement analyst should not get confused and not leave any loop holes in security part of requirement analysis which can be cause of changes in initial deployment of the software. Analyst should not dive in a deep concepts and analysis in one direction, purposed framework can be treated as a complete abstract of the requirement engineering process as well security requirement engineering, our effort is to balancing the requirement analysis in the all part of the system including security issues. As the part of security requirement engineering framework, concepts have been inherited from the existing framework of security requirement engineering [3]. Where in the third stage of the security engineering framework to identify security requirements, seven states of the security as defined in ISO/IEC 13335 standard [18] is mapped to produce security error traceability matrix, as the current edge of technology, in the mobile world Android has captured the market, all the corporate and personal users are relying on the android and think that their data is fully secured. With the help of purposed framework, an argument has done with android to find out security error.

KEYWORDS *- Requirement Engineering, Security System, Android, REPSS- Requirement Engineering Process and Security System*

I. INTRODUCTION

Requirement engineering is the process of finding out, analyzing, documenting and checking the services and constraints is called requirements engineering [6]. Based on the mentioned basic definition of requirement engineering , the process of the same is shown in the purposed framework and integrated with the existing security requirement engineering framework[3] with the major changes , which is done in the third stage of the security requirement engineering process to identify the security requirements, many researchers has approached and purposed many different framework and requirement engineering process in different stages to achieve the final requirement documents as the final output of the requirement engineering process, in the purposed framework, process and stages is simplified and it's easy to compile the whole requirement engineering and security requirement engineering process at a glance, core stages which is involve in the requirement engineering are Requirements elicitation, Requirements analysis, Requirements validation and Requirements management [15], as many researchers has given slightly different stages to complete

requirement engineering process but almost all stages and frameworks suits the common stages in different techniques , in the purposed framework also the core stages of requirement engineering is automatically suits. in this paper in section 2 discussed about brief literature survey, in section 3 explained about existing frameworks used in the purposed system, in section 4 shown the design of purposed framework and discussed stages involved in security requirement engineering (Identify security requirement engineering is explained with the argument and mapped with android) .Result and discussion is mentioned in section 6 and finally future scope is takes place in section 7.

II. BRIEF LITERATURE SURVEY

Requirement engineering also called requirement analysis is the process of determining user expectation for a new or modified product. Requirement analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well skills dealing with people [16]. The requirements for a software-intensive system are strongly influenced by the system context. A sufficient understanding of the context is an essential prerequisite for developing a good requirements specification. The system context comprises a large number of different aspects that are relevant to the system to be developed, such as business processes and workflows, existing hardware and software components, other systems that interact with the system [17]. After completion of the requirement analysis core phases the following are the parameters to check the requirements [15].

- **Validity**: Does the system provide the functions which best support the customer's needs?
- **Consistency**: Are there any requirements conflicts?
- **Completeness**: Are all functions required by the customer included?
- **Realism**: Can the requirements be implemented given available budget and technology
- **Verifiability**: Can the requirements be checked?

Different techniques can be used to complete the above mentioned parameters to check requirement which suits automatically by using purposed integrative framework of REPSS. Security requirement engineering framework is based on constructing a context for the system, representing security requirements as constraints, and developing satisfaction arguments for the security requirements [3].

III. FRAMEWORKS USED IN INTEGRATIVE PURPOSED FRAMEWORK OF REPSS

Based on the conceptual definition [6] of requirement engineering process, we are purposing requirement engineering framework and integrating with security requirement engineering framework [3] This framework is based on constructing a context for the system, representing security requirements as constraints, in this existing framework we have done the major changes which is done in the third phase to identify the security requirements. Security is a property of the system which remains dependable in the face of malice, error, or mischance [3]. In scope of information system, security consists of seven states: confidentiality, integrity, availability, authenticity, accountability, non-repudiation and reliability which is used in the purposed framework to trace security error in the software system and produce final output as security requirement document , which should be included in the final requirement documentation.

IV. DESIGNING OF INTEGRATED REQUIREMENT-BASED FRAMEWORK TO EMPHASIZE SECURITY SYSTEM

As discussed in the abstract in this section designing of purposed frameworks is shown, in the purposed framework, requirement engineering process will be started from the system context in the form of context diagram which will be the base of starting requirement engineering and security engineering process, where initial requirement analysis part can be done as doing questionaries' session with the stakeholders ,in this stage in the security requirement engineering part, security requirements goal should be made and treat entry satisfaction criteria as feasibility study and exit criteria is use to validate the system after customer satisfaction and approval, rest of the stages are clearly shown in the Fig1 and discussed in the further sections.

Stages involved in Security requirement engineering framework

4.1 Identify Functional Requirement & Construct System Context

There should be only one output to represent of the development procedure at this stage

4.2 Initial Requirement

As the part of initial requirement at this stage we should mention overall system requirement including supportive and alternative software and hardware requirements.

4.3 Identify security requirement goal

The security requirement goal is to protect the assets from the harm by setting up the constraint on the functional requirements**.**

There are three general steps required to identify the Security goals **[3]:**

4.3.1 Identify candidate assets

The goal of this step is to find all of the resources in the system context that might have value. In general, assets consist of all of the information resources stored in or accessed by the system-to-be and any tangible resources such as the computers themselves. Assets can be composed of other assets; backup tapes would be a good example

4.3.2 Select management principles

The functions that the system is to provide must be compared to the management principles that the organization wishes to apply. These principles might include separation of duties, separation of function, required audit trails, least privilege (both need to know and need to do), Chinese wall, data protection, no outside connections, and no removable media (not intended to be an exhaustive list). The organization might have already done a harm/risk analysis and developed organization- wide security policies for asset types. Which global policies to apply within the system under consideration must be identified and fed into the next step.

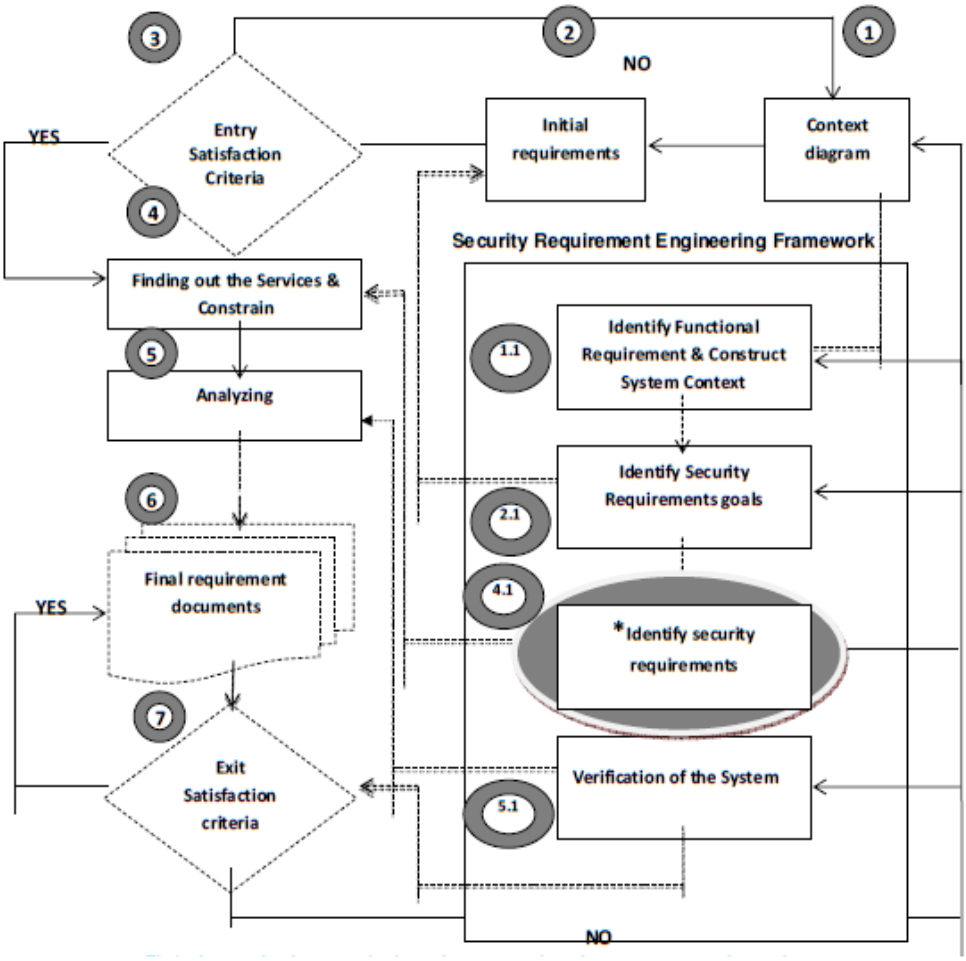

Fig1- Integrative framework of requirement engineering processes and security system

4.3.3 Determine security goals: When developing security goals, one must determine whether a harm analysis must be done for the assets. If the analysis has been done elsewhere (e.g., organization-wide policies) and if the assets are covered by the policies, then a list of security goals is generated by applying the management principles to the assets and business goals of the system. The result is a set of achieve goals with forms similar to "achieve Separation of Duties when paying invoices" or "audit all uses of account information. If the analysis done elsewhere is not considered sufficient, one should do a harm analysis. In general, harm is caused by the negation of the security concerns described in Section 3.1: confidentiality, integrity, availability, and accountability. One asks questions of the form "what harm could come to [insert asset here] from an action violating [insert concern here]?" Answers to these questions are threat descriptions [31], which are represented as tuples of the form {action, asset, harm}. Security goals are constructed by specifying that the action(s) on the asset(s) listed in threat descriptions be prevented. The goals identified from the two analyses (if both are done) must be combined and checked to ensure that they are consistent.

4.4 *Identify Security Requirement: Security requirements are as constraints on functional requirements that are needed to satisfy applicable security goals. To determine the constraints, we must establish which security goals apply to which functional requirements, which means we must know which assets are caught up in fulfilling a particular functional requirement. To support the already proposed framework [3], we are adding the seven stages to identify the security requirement. These seven stages will also suits explore the hidden requirement and finding out the loop holes in the proposed system, in this paper we are mapping our new proposed **Integrative REPSS framework** with the **android security architecture.**

Stages involved in Security requirement engineering framework

4.4.1 Identify Functional Requirement & Construct System Context:
There should be only one output to represent of the development procedure at this stage

4.4.2 Initial Requirement:
As the part of initial requirement at this stage we should mention overall system requirement including supportive and alternative software and hardware requirements.

4.4.3 Identify security requirement goal:
The security requirement goal is to protect the assets from the harm by setting up the constraint on the functional requirements**.**
There are three general steps required to identify the Security goals **[3]:**

4.4.3.1 Identify candidate assets:
The goal of this step is to find all of the resources in the system context that might have value. In general, assets consist of all of the information resources stored in or accessed by the system-to-be and any tangible resources such as the computers themselves. Assets can be composed of other assets; backup tapes would be a good example

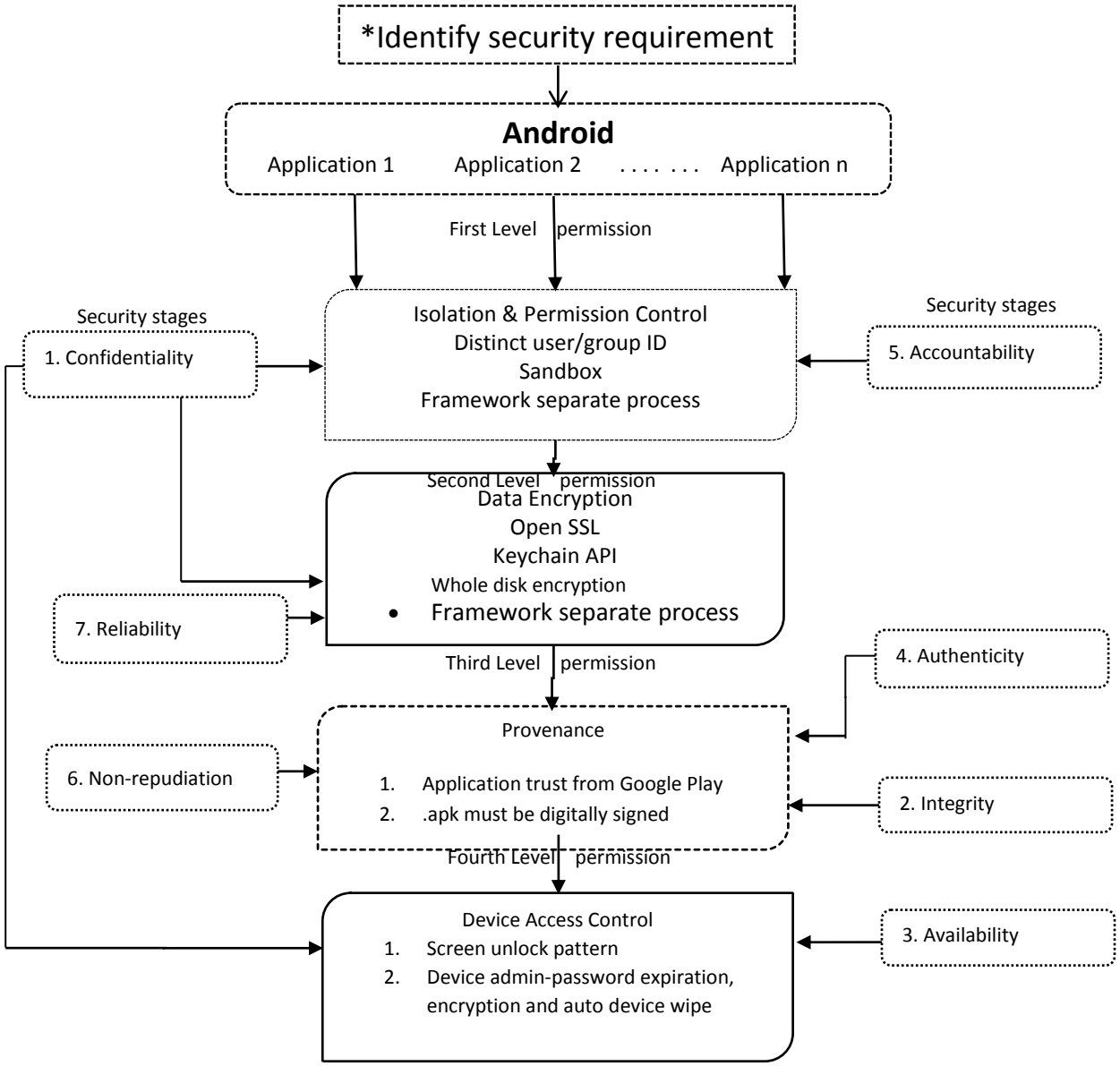

Fig 2- Seven states mapping to identify security requirement with android

4.4.3.2 Select management principles: The functions that the system is to provide must be compared to the management principles that the organization wishes to apply. These principles might include separation of duties, separation of function, required audit trails, least privilege (both need to know and need to do), Chinese wall, data protection, no outside connections, and no removable media (not intended to be an exhaustive list). The organization might have already done

4.5 Seven states mapping to identify security requirement with android as an example

In scope of information system, security consists of seven states:

1. Confidentiality
2. Integrity
3. Availability
4. Authenticity
5. Accountability
6. Non-repudiation
7. Reliability

The seven states which are used in Fig 2 to identify android security requirement are defined in ISO/IEC 13335 standard [5].

Android has become the chief target for malicious hackers. According to several reports, cyber-crooks are targeting the Android operating system since it is open platform and most of mobile users are using Android. Android does not distinguish applications as being trusted or untrusted all applications are created equal [13]. Android allows user to install application on the device unknown sources meaning non-market (Non Google Play) applications.[19] This is the high risk for the phone for any damage or loss of data using these applications. An untrusted application is either sandboxed or severely restricted from accessing any sensitive resource. All social application is able to read /write /edit SMS/MMS, for example Facebook [21]. Text messaging is one of the most popular things people do with their mobile devices nowadays. According to several security firms, Short Message Service (SMS) threats are increasing. Android and other GPS-enabled devices also have access to real-time information about the owner's location. A compromised device can lead to severe financial losses or even social threats. As Android is frequently releasing the new version, different versions have different issues. So, cyber-criminals exploit the security issues found in outdated models and put devices at risk. And in some cases, the same issues might not impact all versions of the software. The open source nature of Android means that Android's infrastructure can be changed arbitrarily, thus making any security infrastructure unreliable.

In the cloud-enabled, highly networked world of modern computing, security is one of the most important facets of proper software engineering. Open source nature of Android is also attracting more and more malware writers. Hence, the growing security problems of Smartphone's are becoming a real concern for users [13]. Service providers need assurance that if sensitive data is released to a Smartphone, it will not be compromised due to the presence of a malware on the phone [13].

The most important thing to understand about security is that it is not a bullet point item. You cannot bolt it on at the end of the development process. You must consciously design security into your app or service from the very beginning, and make it a conscious part of the entire process from design through implementation, testing, and release.

In the software requirement phase, you must determine the nature of the threats to your software and architect your code in such a way that maximizes security. To do this, you should build up a threat model that shows ways in which your software might be attacked.

Ultimately, security (at the application layer) means being aware of how your code uses information and ensuring that it does so safely and responsibly. For example:

If your software is entrusted with personal data that belongs to your users, ensure that your software collects only the information that it requires.

If your software accesses the Internet or files that might have previously been sent to someone over the Internet, ensure that you read that file in a safe manner that does not inadvertently provide a vector for accessing other personal data that may be stored on the user's computer or other mobile device. If your software transmits personal information over the Internet in a safe and secure fashion to prevent unauthorized access to or modification of the data while in transit. If your software provides access to signed data, it is your responsibility to verify those signatures to ensure that the data has not been tampered with.

Table 1 – Security error traceability matrix

SECURITY STATES	SECURITY ERROR
1. confidentiality	Screen unlock, device admin password
2. integrity	Screen unlock, device admin password
3. availability	App without permission installation
4. authenticity	Non digitally signed apps, permission enforcement , application provenance, device access control
5. accountability	Data encryption, permission enforcement, App without permission installation
6. non-repudiation	Application downloaded from internet
7. reliability	Non digitally signed apps, data encryption , permission enforcement, Application downloaded from internet,

The following point shows that, the Android security fails, it the app's are developed and distributed in the market without the application singing of Google playie. Non Google play apps.

With the help of security error traceability matrix as shown on table 1, errors can be find out in the security after drafting the overall requirement analysis, to avoid failure and loosing robustness of our software system, as per the discussed argument we can say that, android still having the lack of security which can be taken out in the similar or any other kind of software system where the security is the main aspects.

4.6 Verification of the System:

It is important to verify that the security requirements are satisfied by the system as described by the context. We propose two-part satisfaction arguments for this verification step: to convince a reader that a system can satisfy the security requirements laid upon it. The first part, the outer argument, consists of a formal argument to prove that the instance of the system under examination satisfies its security requirements, with two important assumptions: that the context is correct and that the implementation will not introduce any conflicting behavior. We recognize that both of these assumptions are very strong and often untrue in practice. Verification that the system can satisfy the requirements cannot ensure the truth of the assumptions, but it does ensure a sound structure for the system that is potentially secure. The second part, the inner argument, consists of structured informal arguments to support the assumptions about system composition and behavior made in the formal argument. Satisfaction arguments assist with identifying security-relevant system properties and determining how inconsistent and implausible assumptions about them affect the security of a system.

V. STAGES INVOLVE IN PURPOSED INTEGRATIVE FRAMEWORK OF REPSS

As shown in **Fig 1** we have integrated security requirement engineering framework with requirement engineering framework. the stages which is involved in requirement engineering framework is very simple and conceptual, and clearly shown in Fig 1, the work flow of the purposed framework is as per the arrow direction and number labeling.

5.1Entry Satisfaction Criteria

Entry criteria will look for need and its checks the feasibility about overall customer and system requirement if entry criteria will fulfilled final requirement document will be produced else refinement procedure must be done as per the work flow direction shown in Fig1.

5.2 Exit Satisfaction Criteria

Exit criteria will look for the satisfaction of all the possible aspects where the security is concern, after consulting the client with the consideration of the external legal policies and issues. If exit criteria is fulfilled final requirement document will be produced else refinement procedure must be done as per the work flow direction shown in Fig1.

VI. RESULT AND DISCUSSION

As the result of this research work , the researchers , academician's can rework on the purposed framework for the further refinement in the area of security and requirement engineering , as discussed in this paper, android architecture is used to identify security requirement engineering , as it's not possible to discuss and map the complete case study for android or any other system within just only one publication, that's the reason only third stage of security requirement engineering is mapped and discussed, it can be extended and shown in the next version of this research work, in the purposed framework many techniques and methodology can be used to complete each phases, purposed framework is well suited to use and treat as a complete abstract about the requirement engineering process including security requirement engineering process.

VII. CONCLUSION AND FUTURE WORK

The purposed framework is simple and robust way to complete requirement engineering process where we have to more emphasize on security system; in this paper we have only shown the android example on third stage to identify the security requirements. As the part of future work and extension of this paper, purposed framework can be mapped with any existing software system to find out the

security loop holes of the software system , or work can be used to get error free security requirement document and over all requirement of any new software development.

REFERENCES

[1]. Security Requirements Engineering: State of the Art andPractice and Challenges, GolnazElahi
[2]. Software & Systems Requirements Engineering: In Practice,by Brian Berenbach, Arnold Rudorfe.
[3]. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 1, JANUARY/FEBRUARY 2008.
[4]. Security Requirements Engineering: State of the Art and Practice and Challenges, Golnaz Elahi www.cs.toronto.edu/~gelahi/DepthPaper.pdf
[5]. ISO/IEC. Management of Information and Communication Technology Security {Part 1: Concepts and Models for Information and Communication Technology Security Management. ISO/IEC 13335,2004
[6]. Embedded software engineering laboratory- http://embedded.korea.ac.kr/esel/se.html
[7]. D.F.C. Brewer and M.J. Nash, "The Chinese Wall Security Policy," Proc. 1989 IEEE Symp. Security and Privacy, pp. 206- 214, 1989.
[8]. S.J. Buckingham Shum, "The Roots of Computer Supported Argument Visualization," Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, P.A. Kirschner, S.J. Buckingham Shum, and C.S. Carr, eds., pp. 3-24, Springer-Verlag, 2003.
[9]. J.E. Burge and D.C. Brown, "An Integrated Approach for Software Design Checking Using Design Rationale," Proc. First Int'l Conf. Design Computing and Cognition, J.S. Gero, ed., pp. 557-576, 2004.
[10]. S. Capkun and J.-P. Hubaux, "Securing Position and Distance Verification in Wireless Networks," Technical Report EPFL/IC/200443, Swiss Federal Inst. of Technology Lausanne, May 2004.
[11]. Australian Technical Standard Order: Airborne Stand-Alone Extended Squitter, Automatic Dependent Surveillance-Broadcast(ADS-B), Transmit Only Equipment," Australian Civil Aviation Safety Authority, Standard ATSO-C1005, CASA, Dec. 2004.
[12]. Common Criteria Sponsoring Organizations, "Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model, Version 3.1 Rev 1," Nat'l Inst. Of Standards and Technology CCMB-2006-09-001, Sept. 2006.
[13]. Beyond Kernel-level Integrity Measurement: Enabling Remote Attestation for the Android Platform - Mohammad Nauman, Sohail Khan, Xinwen Zhangand Jean-Pierre Seifert.
[14]. Google Android: http://www.developer.android.com.
[15]. Ian Somerville 2004, Software Engineering 7th edition. Chapter 7
[16]. http://searchsoftwarequality.techtarget.com/definition/requirements-analysis
[17]. Requirement Engineering Fundamentals, Principles, and Techniques by Pohl, K. Chapter 4, ISBN: 978-3-642-12577-5
[18]. ISO/IEC. Management of Information and Communication Technology Security {Part 1: Concept and Models for Information and Communication Technology Security Management. ISO/IEC 13335,2004.
[19]. Android Developers: http://developer.android.com/index.html
[20]. Deep Drive into Android Security – Aleksandar Gargenta, Marakana Inc.
[21]. Reversing Android Apps – Hacking and cracking Android app is easy – Dreamlab Technologies
[22]. Understanding Android's Security Framework – William Enck and Patrick McDaniel
[23]. https://developer.apple.com/library/ios/#documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html#//apple_ref/doc/uid/TP30000976, Security Overview: About Software Security

AUTHOR'S BIOGRAPHY

Md. Faisal

He has received his graduation degree as B.Sc. in Computer Application (Hon's) from Ranchi University, and post-graduation degree as Master of Computer Application (MCA) from Vishveswaraya Technological University, Belgaum with distinction and he has published many research papers in different international journals in his research area. Currently he works as Lecturer in IT department at Nizwa College of Technology, Oman on contractual basis. His area of research is Software Engineering- Requirement Engineering, Security system and Process Models.

M.Mohideen Abdul Kader Jailani

He has completed his MCA from bhartidarsan university, trichy tamilnadu in and having 18 years of experience in IT Industry as software engineer, team leader and project manager at different level and having 2 years' experience in teaching field, he expertise java, j2EE and Android technologies, his area of research is Software Engineering, Mobile Computing.

Sanjai Gupta

He has received graduation degree as B.Sc. in Mathematics (Hon's) from Lucknow University, and post-graduation degree as Master of Computer Application (MCA) from VBS University, Varanasi with distinction and pursuing his PhD. in computer Science from CMJ University, Megalagay, India. Currently he is working as Lecturer in IT department at Nizwa College of Technology, Oman on contractual basis. His area of research is Software Engineering-Software Development Process.

Mohammed Husain

He has received PhD in Computer Science from Integral University, Lucknow and currently working as Director inM G Institute of Management & Technology, Lucknow, UP, India. He has published many papers in national and international journals.