SECURE IDENTIFICATION ON THE DESIGN AND IMPLEMENTATION OFSECURE NETWORK PROTOCOL

Padala Kavitha,
Research Scholar, Computer Science and Engineering, OPJS University, Churu, Rajasthan.
Dr.Vijay Pal Singh ,
Assistant Professor, Computer Science and Engineering, OPJS University, Churu, Rajasthan.

Abstract

Today's software is more vulnerable to attacks due to increase in complexity, connectivity and extensibility. Securing software is usually considered as a post development activity and not much importance is given to it during the development of software. However the amount of loss that organizations have incurred over the years due to security flaws in software has invited researchers to find out better ways of securing software. In the light of research done by many researchers, this thesis presents how software can be secured by considering security in different phases of software development life cycle. A number of security activities have been identified that are needed to build secure software and it is shown that how these security activities are related with the software development activities of the software development lifecycle. Secure software development processes are critical part of designing secure software. However, it is hard for the various stakeholders to make the decision about which software development process to choose without a comparison between them. Even further, after choosing the process, stakeholders have to decide which methods and techniques to use to fulfill activities required to develop secure software development processes. This is a problem, because there are a number of methods a stakeholder could use to fulfill these activities, but no explicit links between a method and development process.

1. Introduction

Software and software systems have become indispensable tools used in most of the business organizations, educational institutions and in our personal lives as well. However, the software systems always fear danger or risks from the malicious elements due to industrial growth and rapid evolution of technology. The growing Internet connectivity has made the sensitive information more vulnerable to unintentional and unauthorized use. Security metrics and measurements are powerful techniques to SDP. The metrics focus on identifying trust worthiness of security considerations during the development process. Several metric models and security metrics are available to deal with security concerns. Most of the product metrics may be validated while process development stage metrics are still in the nascent phase. Thus, the major contribution towards secured software may be achieved by incorporating security during the software development stages. It requires a thorough understanding of various types of security as well as its technological and management aspect. Systematic consideration of security can be supported by security framework to help the development of secured software product. Moreover, security requirements must be gathered as the first step towards secure development.[16] Security must be incorporated systematically during the software development stages to ensure secured product. Additionally, secured development process must be controlled by help of security metrics.

Some of the popular tools for secured software development include misuse cases, threat modeling, attack trees, attack patterns, source code analyzers, etc. Misuse cases are the business threat modeling tool that describes the process of malicious act against software system. Threat modeling helps to apply structured approach to identify security threats, vulnerabilities and countermeasures. Attack trees are conceptual diagrams to identify the attack methodology.[20] Itis an effective tool for threat analysis and risk assessment. Such tools help in identification of security requirements as well as design related security flaws.

2. Objectives

On the basis of extensive literature review on security aspects and secured software development processes, we designed objectives with some identified challenges. These challenges include understanding the need of security and its integration during the development process. To combat these challenges, we have designed framework, secured software development process and metrics and have been assessed using case studies.

Our first objective is affirmed to identify various security domains for software system, classify these domains as a part of pre-development phase and use the classification for development of different types of software. Classification involves the understanding of various security measures required for secured software systems. At first level, security is classified as hardware security, logical security and security management which is further classified in five more levels.Thus, varied security domains such as hardware security, application software security, system software security, single user system security, peer-to-peer network security, server-based system security, server-based platform security, server-based communication security, web- server security, web communications security, logical data security, physical data security and security management have been identified. Further, we recognized some of the software systems such as static website, dynamic website, web-based enterprise application, intranet and extranet based systems, client/ server based systems, e-commerce, kiosk, cloud, and desktop based systems to identify the security domains involved to secure such systems. Security can be observed as an issue of overall control with software security as one of the major aspects for secured systems. Even with many efforts on security management and network security, security has emerged as a problem of software security.

Our next objective is stated to identify the dominance of security in networked environment. It is evident that there exist many networks such as Internet, client/ server based centralized system etc. with varied security issues. To understand the security issues, we proposed an umbrella of networks based on various dimensions such as size, design, network architecture, organizational scope, computing models and topologies. The network dimensions may be further classified along with the security concerns. Additionally, we tried to identify the dominance of security on the stages of software development process based on varied types of networks. While most of the security breaches are result of insecure software, practitioners still rely more on security configurations and perimeter defense.

The subsequent objective is avowed to identify the technical and management security aspects to secured software development process. Secured software development is a means to software security. It requires systematic security consideration throughout SDP. We developed techno- management view of security that focus on technical and process management aspects of security to be incorporated during Secured Software Development Process (SSDP). The techno- management view of security may help in bridging the gap between process and the management aspects of security. We also introduce generalized Software Product Security (*SPS*) framework for systematic inclusion of security aspects. The framework is a three layered structure consisting of control, security aspects and development layers. These layers may help in systematic inclusion of security during the product development. Further, a mathematical model is formulated to estimate the security concern of some software system developed using *SPS* framework through Security Factor (F_S).

Later objective is established to design secured software development process. Secured development initiates with gathering specific security requirements. We established Software Security Requirements Gathering Instrument (*SSRGI*) that promotes gathering security requirements in detail with the help of the stakeholders. *SSRGI* focuses on gathering Secure Functional Requirements (SFR), Drivers, Functional Security Requirements (FSR), Non-Functional Security Requirements (NFSR), Secure Development Requirements (SDR), and Security Testing Requirements (STR). When *SSRGI* is integrated with Software Requirements Specification (SRS) document, it may promote the systematic gathering of security requirements along with the functional requirements. Since software developers might unknowingly inject flaws during software development process, SSDP has been designed to address security issues during each development phase. It may support the development team in security considerations throughout the development stages. It shall ensure that the updates and patches do not add security weaknesses during SDP.

Our last objective is concerned with identification of security metrics for each of the software development stages. Metrics serve as a basis for analyzing security improvements by investigating data regarding measurements and metrics such as number of security requirements gathered, number of design flaws related to security, mean time between security incidents etc. throughout the development. It also analyzes product implementation by measuring the security parameters such as failure in audit capturing mechanisms, number of known vulnerability incidences etc. for improving performance and accountability. In this respect, we analyzed currently available metrics for software development stages. It has been observed that late security assessment cannot help in correcting the security issues of early stages. Moreover, the process metrics do not provide sufficient details to analyze security. We have developed security metrics on the basis of security issues of the development phases. Further, a methodology to identify the effectiveness factors to analyze and judge the security efforts have been built. The metrics may support the development team in evaluating and monitoring the security efforts throughout the development process.

3. Work Plan & Methodology

3.1. Proposed Hierarchical Classification

To understand the various dimensions of security, we classify security hierarchically at six levelsranging from Level 1 to Level 6. Level 1 defines the major types of security required by the organization for implementing secured software systems. It categorizes security into further three classes viz. Hardware security, Logical security and Security management as shown in Fig. 2. Level 2 to Level 6 define the subtypes for achieving parent level security. In following section, we discuss the proposed hierarchical classification of security in IT environment:

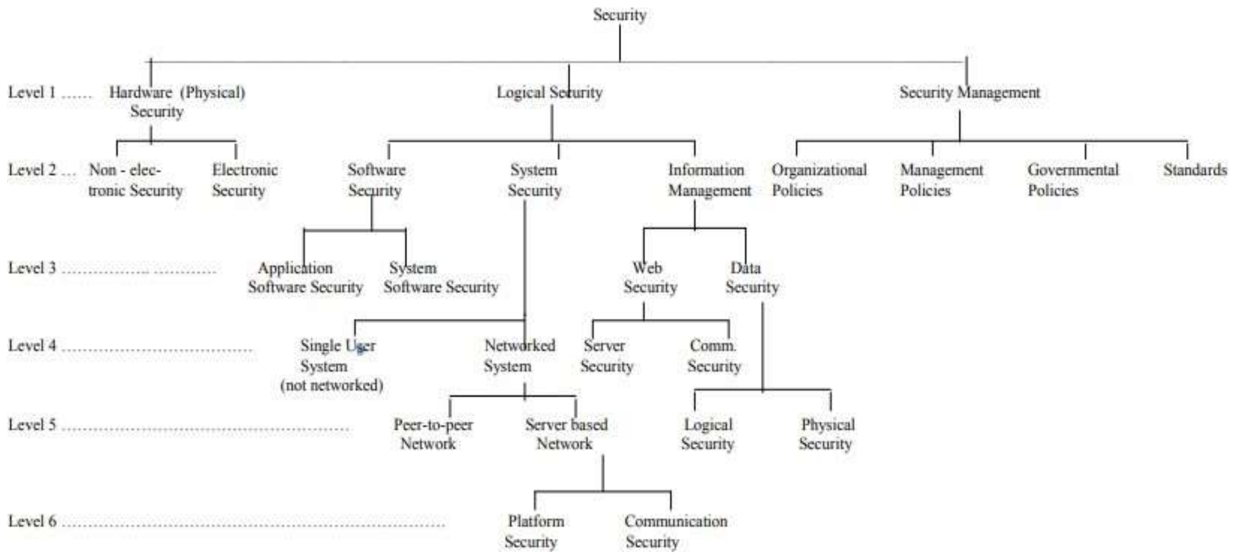
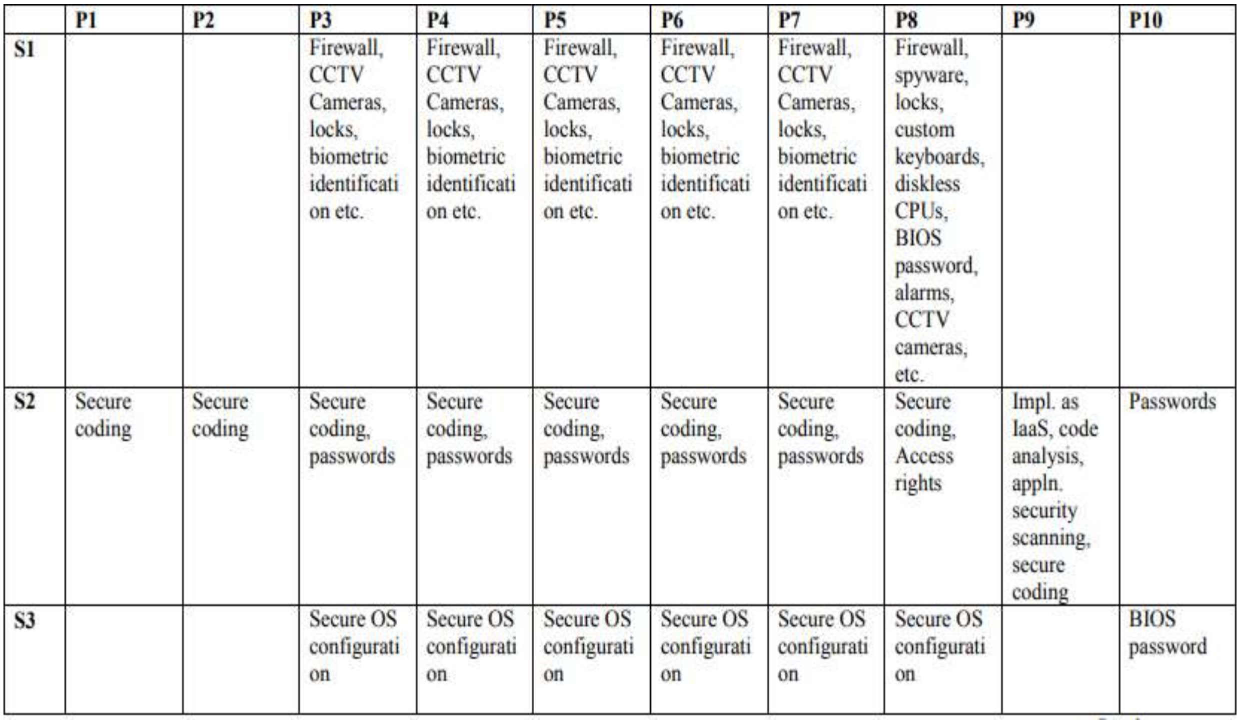


Fig 1: Proposed Hierarchical Classification of Security4.Software

Systems And Security Issues

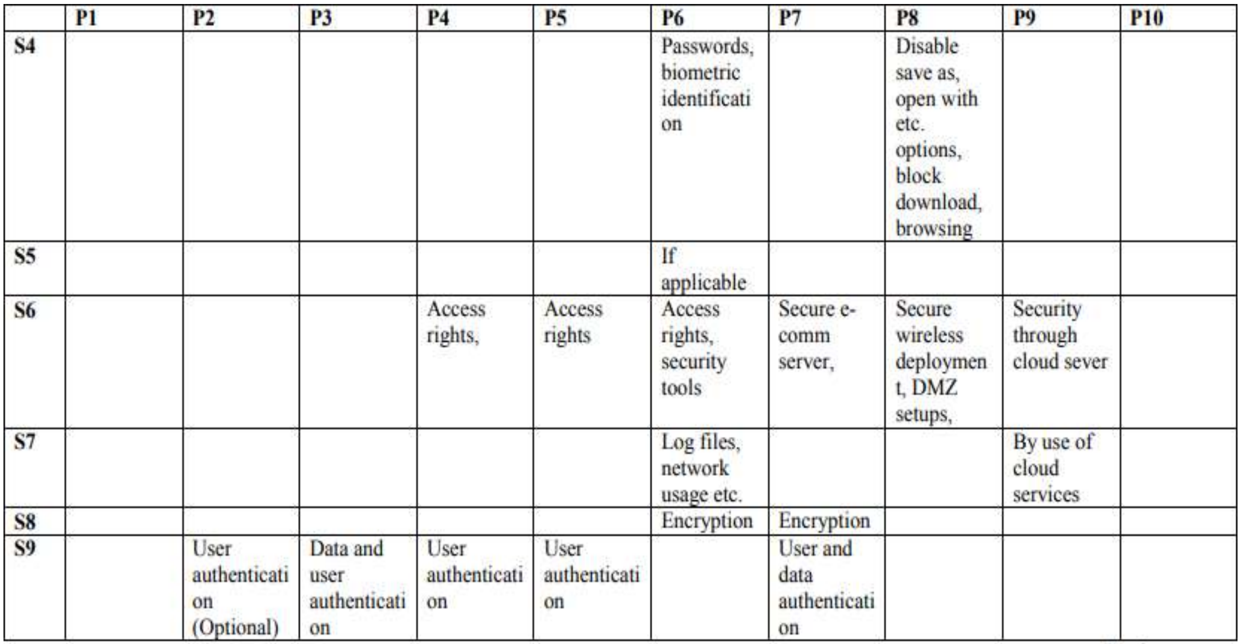
The software systems are developed based on organizational scope and the technology used for implementation. Depending on organizational scope, projects can be based on Internet, Intranet, and Extranet. Based on the underlying architecture, software system can use client/ server, web technology and cloud based technology. The websites can be static or dynamic depending on the contents of the website. In the same context, we have selected 10 types of software systems namely; static website (P1), dynamic website (P2), web-based enterprise application (P3), Intranet based systems (P4), Extranet based systems (P5), client/ server based systems (P6), e- commerce based systems (P7), Kiosk based systems (P8), cloud based systems (P9) and desktop

based systems (P10). The security required for the various types of software systems arediscussed below.



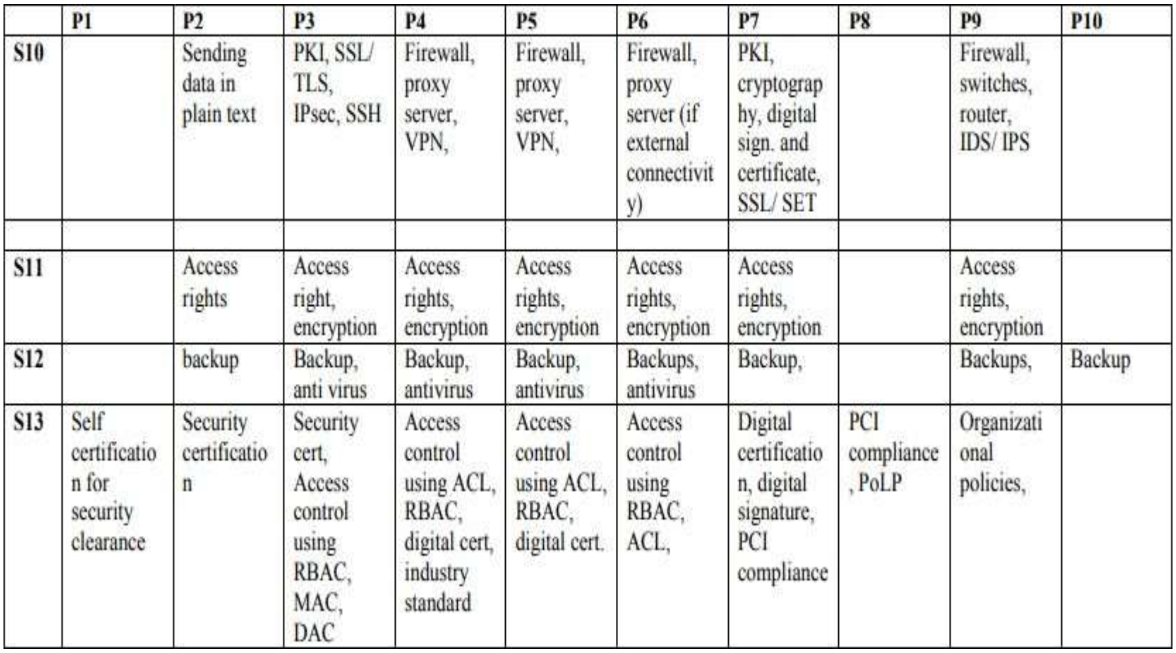
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
S1			Firewall, CCTV Cameras, locks, biometric identification etc.	Firewall, CCTV Cameras, locks, biometric identification etc.	Firewall, CCTV Cameras, locks, biometric identification etc.	Firewall, CCTV Cameras, locks, biometric identification etc.	Firewall, CCTV Cameras, locks, biometric identification etc.	Firewall, spyware, locks, custom keyboards, diskless CPUs, BIOS password, alarms, CCTV cameras, etc.		
S2	Secure coding	Secure coding	Secure coding, passwords	Secure coding, passwords	Secure coding, passwords	Secure coding, passwords	Secure coding, passwords	Secure coding, Access rights	Impl. as IaaS, code analysis, appln. security scanning, secure coding	Passwords
S3			Secure OS configuration	Secure OS configuration	Secure OS configuration	Secure OS configuration	Secure OS configuration	Secure OS configuration		BIOS password

Table 1 (a): Categorization of Security Issues



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
S4						Passwords, biometric identification		Disable save as, open with etc. options, block download, browsing		
S5						If applicable				
S6				Access rights,	Access rights	Access rights, security tools	Secure e-comm server,	Secure wireless deployment, DMZ setups,	Security through cloud sever	
S7						Log files, network usage etc.			By use of cloud services	
S8						Encryption	Encryption			
S9		User authentication (Optional)	Data and user authentication	User authentication	User authentication		User and data authentication			

Table 1 (b): Categorization of Security Issues



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
S10		Sending data in plain text	PKI, SSL/ TLS, IPsec, SSH	Firewall, proxy server, VPN,	Firewall, proxy server, VPN,	Firewall, proxy server (if external connectivity)	PKI, cryptography, digital sign. and certificate, SSL/ SET		Firewall, switches, router, IDS/ IPS	
S11		Access rights	Access right, encryption	Access rights, encryption	Access rights, encryption	Access rights, encryption	Access rights, encryption		Access rights, encryption	
S12		backup	Backup, anti virus	Backup, antivirus	Backup, antivirus	Backups, antivirus	Backup,		Backups,	Backup
S13	Self certification for security clearance	Security certification	Security cert, Access control using RBAC, MAC, DAC	Access control using ACL, RBAC, digital cert, industry standard	Access control using ACL, RBAC, digital cert.	Access control using RBAC, ACL,	Digital certification, digital signature, PCI compliance	PCI compliance, PoLP	Organizational policies,	

Table 1 (c): Categorization of Security Issues

In our study, we have analyzed the type of security required for the various software systems. Asseen from Table 1, application software security may be considered for all software systems but the main security concerns for application software security are mentioned as passwords and secure coding. It has been mentioned that 75% of breaches are mainly due to vulnerabilities in the applications although most of the security efforts lie with the networks. It has also been argued that considering security throughout the development process of the software can substantially improve the security of the software.

Security management is considered to be next important aspect for securing the software systems. Except for desktop systems, security management is required for all the types of software systems. Most of the software systems are considering organizational policies to secure the systems. Physical data security is important to protect the data against the loss due to data corruption. Logical data security is required in web-based enterprise application, intranet, extranet, client/ server, e-commerce, dynamic website, and cloud based system. According to Symantic, majority of IP theft is committed by the technical insiders who are authorized to access the data. Hence, the management needs to focus on access control matrix and role based access control mechanisms. Web communication security is important for dynamic website, web-based enterprise applications, Internet, Intranet, Extranet and cloud based systems.

The hierarchical classification of security and its importance in various types of software can be beneficial to management, developers as well as system and network administrators in addressing the various security issues regarding the type of application being developed. The usefulness of the classification to the various stakeholders is discussed below:

4.1. Management

The security management personnel may be able to focus on the key security aspects required forthe implementation and maintenance of the system. Classification shall also help to focus on the perimeter defense as well as appropriate security policies, procedures, compliances and risk management. The management may also promote the training of the system personnel to managesecurity of the system.

4.2.Development Team

The classification guides the system developers regarding the areas to focus while implementing security. The developers mainly have to focus on the software development process, taking into considerations the security features of platform, hardware, operating system, networking, and communication. It shall guide the team in testing the various aspects of system during development and after implementation.

4.3.Administrators

The system administrators can focus on securing the data logically and physically. Based on the current security breaches, the administrators can secure the systems by updating the system software configurations, server configurations, and communications security protocols. The systems can also be secured by access control mechanisms for application software, data, and networks. Further, the systems can be secured by means of hardware control that must be updated as required.

Security is a multifaceted problem in the networked environment and therefore requires multiple solutions. Moreover, no matter how good the security devices are, they all are composed of imperfect software. Security is the problem of the applications as well the software and hardware required for securing the system. Mostly, problems arise not only from unexpected interactions between security software and application software, but also due to human operational errors which may be deliberate or unintentional. Thus, security of a system must take into considerationthe security features from the various security domains such as hardware security, application software security, communications security, organizational policies, etc. The classification will help the various stakeholders in systematic consideration of security for different kinds of software systems.

5. Conclusion

In this chapter, we presented hierarchical classification of security in an attempt to understand the different security dimensions while implementing software system. At Level 1, Security has been classified as physical security, logical security and security management based on major types of security required by the organization for implementing secured software systems. Security is further classified at 5 more levels ranging from Level 2 to Level 6. Based on the classification, various security domains are identified required for secure implementation of software systems. Further, we discussed various software projects such as web-based applications, dynamic website, client/ server based, kiosk based software systems etc. Based on the analysis, it has been recognized that the application software security is the key for secured software systems although security has to be treated as an integral part of the overall system design. The classification indicates the need for considering various security domains during the development of secure software systems. The hierarchical classification may help the managers, development team and administrators to understand and consider security during development ofsoftware in a better way.

References

1. Allen, J. H., Barnum, S., Ellison, R. J., McGraw, G., Mead, N.R. (eds), Software SecurityEngineering: A Guide for Project Managers, Addison Wesley Professional, 2018.

2. Abbott, R. P., Chin, J. S., Donnelley, J. E., Konigsford, W. L., Tokubo, S., and Webb, D.A., ―Security Analysis and Enhancements of Computer Operating Systems‖, NBSIR 76-104I, Institute for Computer Sciences and Technology, National Bureau of Standards, Bisbey, Apr. 1976.

3. Arora, G. and Arora, D., ―Web Based Client Server Technology – A Three Tier Architecture‖, [Online] Available: gagnesharora.com/ieee2.pdf

4. Aime, M.D., Atzeni, A., and Pomi, P.C., ―The Risks with Security Metrics‖, QoP'08, ACM, 2008, pp. 65-69.

5. Antonelli, C.J., Doster, W.A., and Honeyman, P., ―Access Control in a Workstation- Based Distributed Computing Environment‖, CITI Technical Report 90−2, 1990, [Online] Available: http://www.citi.umich.edu/techreports/reports/citi-tr-90-2.pdf

6. Alshammari, B., Fidge, C., and Corney, D., ―Security Metrics for Object-OrientedDesigns‖, IEEE Computer Society, 2019, pp. 55-64.

7. Araujo, R. and Gupta, S., ―Design Authorization Systems Using SecureUML‖,Foundstone Professional Services, Whitepaper, 2015.

8. Allen, J. H., ―Building a Practical Framework for Enterprise-wide Security Management‖, In Secure IT Conference Networked Systems Survivability, Apr. 28, 2004.

9. Allen, J., ―Measuring Software Security,‖ CERT Research Annual Report, Software Engineering Institute, Carnegie Mellon University, 2009, pp. 64-65.

10. Alhazmi, O.H., Malaiya, Y.K., and Ray, I., ―Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems‖, Computers and Security Journal, Vol. 26, No. 3, May 2007, pp. 219-228.

11. Araujo, R., ―Security Requirements Engineering: A Road Map.‖ Security Feature, Jul. 2007, [Online] Available: http://www.softwaremag.com/ l.cfm?doc=1067-7/2007.

12. Aslam, T., ―A taxonomy of Security Faults in the Unix Operating System‖, M.S. Thesis, Purdue University, 1995.

13. Avinum, R., ―Concurrent Hardware/Software Development Platforms Speed System Integration and Bring-Up‖, Cadence Design Systems, Inc., 2017.

14. Alvi, A. K., and Zulkernine, M., ―A Natural Classification Scheme for Software Security Patterns‖, In Dependable, Autonomic and Secure Computing (DASC '11), IEEE,2011, pp. 113-120.

15. Buyens, K., Gr´egoire, J., Win, B.D., Scandariato, R., and Joosen, W., ―Similarities and Differences Between CLASP, SDL, and Touchpoints: the Activity-Matrix‖, Report CW501, K.U.Leuven, Oct. 2007.

16. Boehm, B.W., Brown, J. R., Kaspar, H., Lipow, M. MacLeod, F.J. and Merritt, M.J. ―Characteristics of Software Quality‖, TRW Series of Software Technologies, 1, North-Holland, Amsterdam, 1978.