A Novel Framework for Security Requirement Prioritization

Shalini Sharma
Department of CSE
Delhi Technological University (DTU), Delhi, India

Ajit Singh Malik
Department of Management
Birmingham City University, Birmingham, UK

ABSTRACT
Security Requirements prioritization is one of the important Processes in the Software engineering, which aims at identifying and prioritizing the most crucial security requirements for the software project. In order to systematically perform this activity, many approaches have been introduced so far. Despite of the functionalities offered, these techniques have got certain pitfalls imbibed in them such as inefficient and inappropriate requirement gathering prioritization and hike in the specified project budget that leads to degradation in the software quality and security. So there is an imperative need for the efficient solution to overcome them. Thus In this paper, we have proposed a new methodology to prioritize the software security requirements generation process. This methodology improves the security in software applications of the business environment by gathering the properly processed requirements, identifying the vulnerabilities and their corresponding threats. Thus, it leads to the reduction in the estimated budget of the software application along with the security implication.

Keywords
Security Requirements, Threats, Vulnerabilities, Assets, Prioritization, Security.

1. INTRODUCTION
Software applications have become frequently ubiquitous, heterogeneous and susceptible. As reported in CERT [26], a lot of threats are being directed towards software community. These threats are too dangerous and harmful in nature. We need to extend such systems which would be threat–free. Thus, our major focus is to identify all the vital threats and prioritize them. Insecure software system increases the cost, time and may lead to the loss of customer's confidential data, which result into customer's dissatisfaction about using software.

To develop secure software, a lot of investments have to be done. In order to ensure safe and effective capital investment, we need to develop a threat-free system. If we don't develop a secure system then losses will be imparted on the software industry and hence, other industries will also be affected [31].

In general, any software can easily be targeted by viruses, outside attackers, application threats and intruders, etc [31]. If redundant applications are embedded within our main software, then its efficiency will not only be degraded but may even be vanished. Consequently, its reliability and performance gets deteriorated. To avoid failure of software applications, most of the software engineers generate software security requirements. This stage identifies, captures all the major threats to the system and then constructs a firewall that can defend the system from all such threats. System Requirements have been developed by software generators to avoid failure of cost potential system [11, 12, 14, 23] which could get priority to be detected, sustained and further applied. Same principle may be applied to derive security mechanism. Our study shows that current software-development processes implement security events during design phase. In order to ensure safety of the software system, we must detect and collect the security requirements during early phase of SDLC. It thus intervenes with the efficient implementation by specifying the security related constraints. Contrastingly, the performance may be improved if we opted for some other mechanisms [10].

Security requirements are in sync along with functional requirements and hence are required to be captured along with. Security requirements should be accurate, adequate, absolute and non- conflicting with other requirements. Once they have been explicitly specified, they can then be implemented and maintained [33].

These requirements are associated with assets that must be protected and managed. Security requirements should be properly completed. If the assets are damaged, then it will be highly critical for the system and moreover, the value of the assets will also be increased. Some security systems (power point, Banking, Army plan, Science project) are more critical and their security is vital, because even a single threat can cause the complete failure of the system which may force the whole process to start from the scratch.

We aim to develop a well-defined process for security prioritization. Our study shows that we need to use the requirement prioritization which will prioritize each security requirement according to asset value, vulnerability and threat to decrease the cost and hence reducing the application development time.

2. RELATED WORK
There are numbered of proposals for eliciting security requirements using techniques like an abuse case [4], misuse case [1, 2, 21, 22] common criteria [3, 24] or attack trees [6]. These had been implemented using templates [19] but do not signify integrated with the traditional requirement engineering process. There have been proposed certain methods of modeling languages and methodologies like secure tropos, extension of tropos methodology [29]. Intentional anti model extension of KAOS methodology with security requirement oriented to construct [30]. However, these proposals do not account for the cost effectiveness to ensure security.

According to Fire Smith [7], the security requirements had been defined as being a necessary and fundamental component of the system that rendered detailed specification

of uncharacterized system behavior. He also distinguished security requirements had been discriminated from security related architectural constraints so that true security requirements can be figured out by the requirement engineer. Different types of security requirements as proposed by Firesmith [7] are given in [25]. Many vulnerability and security assessment tools have been developed [17, 19]. However, nearly all these efforts are collocated in finding vulnerabilities in the software that have been developed and deployed. Moreover, most of these tools [17] figured out security breaches in network-based systems. Clear point [20] is one of the methodologies to assess the overall security state of a software system that also takes into account the organizational security policies.

Various Risk Management approaches are AHP method [5], Impact validation method [7], cost-value approach [8] and OCTAVE [28]. DREAD [26] has been signified, which enhance security measurement levels in accordance with applied risk levels. An approach has been proposed recently by N.Mayer in order to integrate security constructs and risk management techniques for information system development method. According to AHP method, Karlsson and Ryan [8] proposed cost-value approach. It suggests that the comparison of requirement pairs is based on their significance and applicable costs. The percentage proportion in accordance with each requirement with total value and total cost related to all requirements are manipulated [3].

Karlsson and Regnell [8] have defined the proposed Ordinal cost-value approach. They categorized requirements into three groups in accordance with their value to customers and their implementation costs.

Impact validation method [7] suggested that every proposed requirement has an influence over acquired high level project targets. The most influencing requirement becomes the most critical one, and accordingly, the others may be given less priority.

Octave model [32] made the organizations understand, assess and address their information security risks from the organization's point of view. The analysis team had been formed of the people from operational, business and IT department and deals with addressing the security needs of the organization so that the risk of critical assets had been reduced. There had been no consideration of which security prioritization strategy to be applied for prioritizing the vulnerability and security requirements. OWASP does not anticipate [13] that OCTAVE will be used widely by application designers or developers, because it fails to take threat risk modeling into consideration, which is useful during all stages of development by all participants, to reduce the overall risk of an application becoming vulnerable to attack.

Security Quantification Methodology [9] provided the proper steps to prioritize the vulnerability based upon security requirements. But this model lags to identify the security requirements and their counterpart vulnerabilities. In the absence of proper values, the project might be executed but failure triggered [15].

The proposed framework is hybrid of Octave and Security Quantitative model. First, we are going to address Octave model to identify the security requirement, assets, threats and organizational vulnerability. Then we have applied the Security Quantification Methodology to address vulnerability and security prioritization [32][16][9].

3. PROPOSED FRAMEWORK FOR EARLY SECURITY REQUIREMENT PRIORITIZATION

The block diagram (see fig1) given below describes the major components and work flow of Hybrid Security Requirement Prioritization framework.

3.1 Workshop-based approach

The Workshop–based approach [7, 32] is used for gathering information and making decisions for organization. Knowledge elicitation workshops are facilitated by the analysis team of senior management, operational area management, project manager and staff. The purpose of the knowledge elicitation workshops is to identify the following information which could differ from organization to organization, depending on the perspective of business:

- Important assets and their relative values.
- Perceived threats to the assets.
- Security requirements.
- Organizational vulnerabilities.

3.2 Relationship between vulnerability and error

We need to gather all the vulnerabilities and respective error, which can cause these vulnerabilities in a particular software system [9].

3.3 Relationship between vulnerability and Security Requirement

The next step is to identify a relationship between security requirements and vulnerability so that after ranking these, we can remove all the vulnerabilities from the security requirements [28].

3.4 Relationship between assets and vulnerability

The relationship between assets and vulnerability [9] defines the impact of the vulnerability over the assets and then mentions the potential damage cost ((DANVXOY ((where DAiVjOk is the damage caused to the ith asset by the kth occurrence of the jth vulnerability))) to each asset caused by vulnerability occurrence. Then it defines the implementation cost.

3.5 Calculate total damage

Here we have been computing the total damage cost (TDVXOY) [9] which may be caused by vulnerability as per the assets. We need to get the total of all the assets damage values as per vulnerability occurrence.

$$TDV_X O_Y = \sum_{n=1}^{m} DA_n V_X O_Y \qquad (1)$$

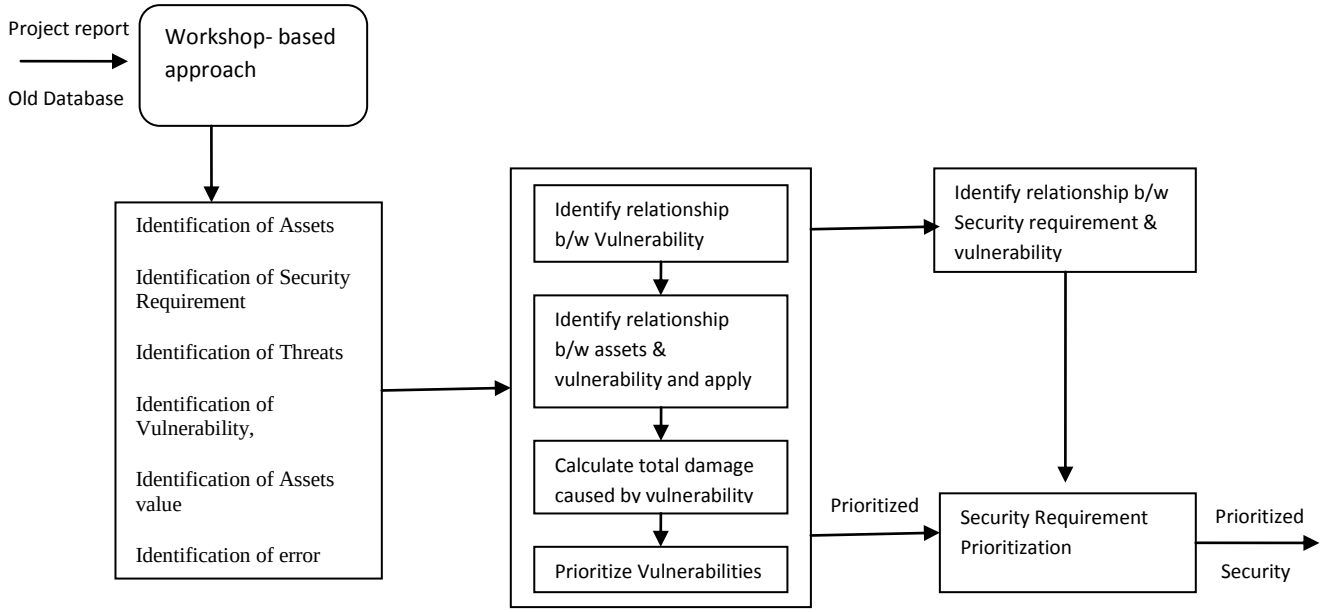


Fig 1: Hybrid Security Requirement Prioritization Block Diagram

3.6 Vulnerability Prioritization

Vulnerability prioritization [9] is based on the difference between the total possible damage cost (TDVXOY) to each asset and the implementation cost (CVXOY) to remove errors

which causing a vulnerability occurrence.

$$PV_X O_Y = TDV_X O_Y - CV_X O_Y \quad (2)$$

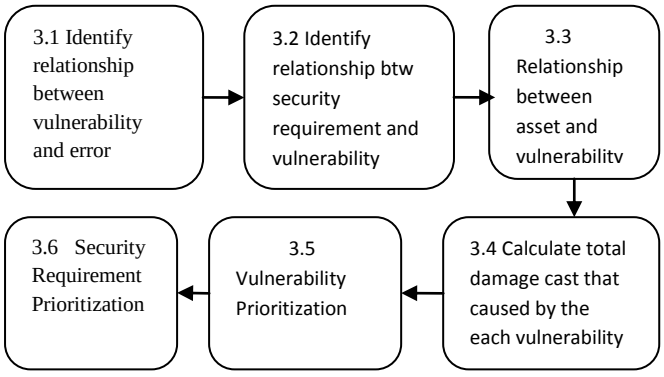


Fig 2: Modified Security Quantification Methodology

3.7 Security Requirement Prioritization

Security requirements have been prioritized according to the value of vulnerability prioritization. First, we have to identify a relationship between security requirement and vulnerabilities. According to that relationship the value of vulnerability priority will be assigned to corresponding security requirement.

Case1 – It is the sum of the priority values of all vulnerabilities, if corresponding to one security requirement there are more than one vulnerabilities.

Case2 – If there is only vulnerability corresponding to security requirement than what so ever is the value of vulnerability priority that will be assigned to the security requirement.

4. A HYPOTHETICAL SECURITY REQUIREMENT PRIORITIZATION ANALYSIS

In this section, we will implement the proposed framework using a case study of "Online Banking system (OBS)". In this application, the customer can create an account electronically through internet and can get the account id on their email-id. The customer can perform online transactions through electronic payment gateways and can also inquire about account details viz canceled transaction details, etc.

4.1 Implementation and Results

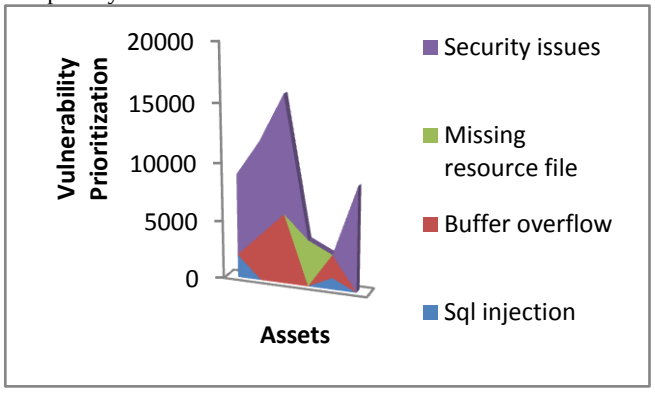
In order to elicit the software requirements for OBS we have used the Workshop–based approach. After applying this, we have collected the number of Assets, Vulnerabilities, Threats and Security Requirements and defined the relationship between each other.

We have summarized six assets and their corresponding vulnerabilities in the following table.

Table 1: Measure of Relationship between assets and vulnerability occurrences

Assets	**Sql injection Vulnerability**	**Buffer overflow Vulnerability**	**Missing resource file Vulnerability**	**Security issues Vulnerability**
User Login IP	$DA_uV_{sql}O_1$ =2.000, $CV_{sql}O_1$=1,000			$DA_uV_{si}O_1$ =7.000, $CV_{si}O_1$=1,000
Credit card IP		DAci$V_{bov}O_1$=4.000,$CV_{bov}O_1$ =1,000		$DA_{ci}V_{si}O$ =8.000, $CV_{si}O_1$=1,000
Debit card IP		$DA_{Dc}V_{bov}O_1$=6.000,$CV_{bov}O_1$=1,000		$DA_{Dc}V_{si}O_1$=10.000, $CV_{si}O_1$=1,000
Communication page			DAccVms O_1=4.000 CVmsO$_1$=1000	
Cancellation IP	DAciVsql O_1= 1.000, CVsqlO_1=1,000	DACiVbovO_1 =2.000,CVbov O_1=1,000		
Make payment IP				DAMpVsi O_1 =9.000 CVsiO_1=1,000
TDVXOY	3,000	12,000	3,000	34,000

The Total damage cost per assets for the OBS is (3,000, 12, 000, 3000, 34000). Thus Information content with the highest value of total damage cost is most valuable and it has got the first priority.



Graph 1: Measure of Vulnerability Prioritization

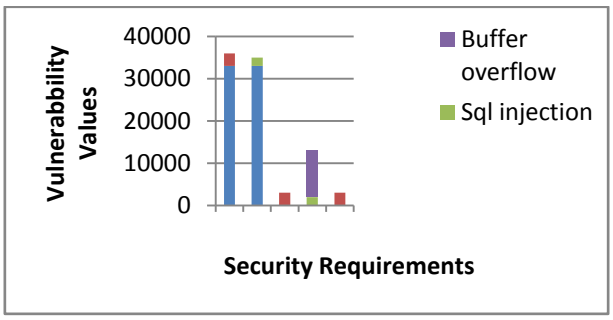
Graph1 shows the prioritization of different vulnerabilities. As per result, after applying the equation2, Vulnerability Prioritization has been ordered. Security issues vulnerability has first priority then buffer overflow vulnerability has second priority and so on.

Finally, we have arrived at our real task that is the calculation of priority of security requirement. We calculate the priority of security requirement just from the value of vulnerability priority.

Table 2: Measure of Security Requirement Prioritization

Security Requirement	**Assets Name**	**Vulnerabilities**	**Vulnerability prioritization value**	**Security Requirement Prioritization**
Authorization Requirement	Ussr login page, Credit card info, Debit card info, Communication channel, Make payment page	Security issues Missing Resources	33,000 3,000	36,000
Authentication Requirement	Communication channel, User login page,Cancellation page	Sql injection Security issues	2,000 33,000	35,000
Privacy Requirement	Communication channel	Missing Resource	3,000	3,000
Integrity Requirement	Credit card info page Debit card info page Cancellation page	Sql injection Buffer overflow	2,000 11,000	13,000
Identification Requirement	User login page Communication channel page	Missing Resources	3,000	3,000

Table 2 shows the different requirements and the values of their associated vulnerabilities, that we used for the process of prioritization of security requirements.



Graph 2: Measure of Security Requirement Prioritization

Graph2 shows the prioritization of different security requirements viz authorization, authentication, integrity, privacy and identification with respect to the occurrence of different vulnerabilities such as buffer overflow, sql injection, missing resources and security issues. It has come to notice that the Authorization Requirement has got the first priority, Authentication Requirement has second, Integrity Requirement comes at Third level; Privacy and Identification Requirement have fourth priority.

5. CONCLUSION

In this paper, we have presented the techniques for discovering security requirements along with functional and non-functional requirements. In addition, we have shown a method first to prioritize the vulnerabilities and based on that we prioritize the security requirements. We have illustrated this method with the help of example. Further complexities in these techniques are under processing. We are also implementing a system-based tool to incorporate these steps. In the future, try to be extend this work to incorporate the security characteristics, besides the CAME tool MERU [9 will be initiated in the construction of method, which includes the security engineering.

6. REFERENCES

[1] Alexander IF, "Modeling the interplay of conflicting goals with use and misuse cases". In: Proceedings of the 8th international workshop on requirements engineering: foundation for software quality (REFSQ'02), Essen, Germany, 2002.

[2] Alexander IF, "Misuse cases, use cases with hostile intent". IEEE Software, 2003, pp. 58– 66.

[3] Common criteria for information technology security evaluation. Technical report CCIMB 99–031, Common Criteria Implementation Board, 1999.

[4] John Mc Dermott, Chris Fox, "Using abuse case models for security requirements analysis." Department of Computer Science, James Madison University, 1999.

[5] KARLSSON, J. and RYAN, K. "A Cost- Value Approach for Prioritizing Requirements", IEEE Software 14 (5), pp. 67–74, 1997.

[6] Robert J. Ellison, "Attack Trees", Software Engineering Institute, Carnegie Mellon University, 2005.

[7] Donald G. Firesmith, "Engineering Security Requirements", Journal of object technology, 2003, vol 2, no.1, pp.53-68.

[8] KARLSSON, L. and REGNELL, B. "Comparing Ordinal and Ratio Scale Data in Requirements Prioritisation", Workshop on Comparative Evaluation in Requirements Engineering, 2005.

[9] Muhammad Umair Ahmed Khan and Mohammad Zulkernine, "Quantifying Security in Secure Software Development Phases", Annual IEEE International Computer Software and Applications ConferenceIEEE, 2008.

[10] GILB, K. "Evo - Evolutionary Project Management & Product Development". Book, 2006.

[11] Johnson, J. "Chaos: The Dollar Drain of IT Project Failures," Application Development Trends, January 1995, pp. 41-47.

[12] Lubars M., Potts C., Richer C., "A review of the state of the practice in requirements modeling", Proc. IEEE Symp. Requirements Engineering, San diego 1993

[13] OWASP, https://www.owasp.org/index.php/Threat_Risk_Modeling#OCTAVE.

[14] Karen Mc Graw, Karan Harbison, "User Centered Requirements, The scenario based", 1997.

[15] VILHELM VERENDEL," Some Problems in Quantified Security", CHALMERS UNIVERSITY OF TECHNOLOGY, Göteborg, Sweden 2010.

[16] EBIOS-Expression of need and identification of security objectives, DCSSI, France, February, 2004.

[17] Nessus. Configuring Nessus to perform local security checks on Unix hosts. http://nessus.org/documentation /index. php, Last Accessed 30-01-2008.

[18] STAT Scanner. http://www.lumension.com, Last Accessed 30-01-2008.

[19] F. Guo, Y. Yu, and T. Chiueh, "Automated and Safe Vulnerability Assessment", In Proc. of the 21st Annual Computer Security Applications Conference, Tucson, AZ, USA, 2005, pp. 150-159.

[20] Clearpoint. http://www.clearpointmetrics.com Last Accessed 30-01-2008.

[21] Sindre G, Opdahl AL, "Eliciting security requirements by misuse cases". In proceeding 37th Conference Techniques of Object-Oriented Languages and Systems, TOOLS Pacific 2000, pp 120-131.

[22]Sindre G, Opdahl AL, "Eliciting security requirements with misuse cases". Requirements Engineering 10, Springer-Verlag London Ltd, January 2005, pp. 34-44.

[23] The Standish group, Chaos. Standish Group Internal Report,1995,http://www/standishgroup.com/chaos.html.

[24] M. Ware, J. Bowles, C. Eastman, "Using the common criteria to Elicit security Requirements with use cases", 2006 IEEE Computer Society.

[25] Agarwal A, Gupta D, "Security Requirement Elicitation Using View Points for online System". 2008 IEEE Computer Society.

[26] CERT/Internet security vulnerabilities. Available Online: http://www.cert.org.

[27] N. Mayer, P. Heymans, R. Matulevičius "Design of a Modelling Language for Information System Security Risk Management", In Proceedings of the First International Conference RCIS – 2007.

[28] Alberts, Christopher and Dorofee, Audrey. OCTAVE Method Implementation Guide v2.0. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.cert.org/octave.

[29] a)Paolo Giorgini, G.Manson,Haralambos Mouratidis. I.Philip, "A Natural Extension of Tropos Methodolog y for Modelling Security". In the workshop on Agent - oriented methodologies, at OOPSLA 2002.

b)Paolo Giorgini, G.Manson, Haralambos Mouratidis. I.Philip, "Modelling Secure Multi agent System". AAMAS- 2003.

[30] A.van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models", Proceedings of the 26th International Conference on software engineering (ICSE'04), IEEE Computer Society, Washington DC USA, 2004, pp. 148-157.

[31] Tom Olzak" A Practical Approach to Threat Modeling", March 2006.

[32] Alberts, Christopher and Dorofee, Audrey. "OCTAVE Method Implementation Guide" v2.0. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.cert.org/octave.

[33] Yngve Espelid" Practices in Software Security", University of Bergen, Norway,2008.