PROPOSING SECURITY REQUIREMENT PRIORITIZATION FRAMEWORK

Aayush Gulati[1], Shalini Sharma[2] and Parshotam Mehmi[3]

[1]Department of Computer Science and Engineering, Lovely Professional University(LPU), Punjab, India
aayush_23kap@yahoo.co.in
[2]Department of Computer Science and Engineering Delhi Technological University (DTU), Delhi, India
shalinisharma13@gmail.com
[3]Department of Computer Science and Engineering, Lovely Professional University(LPU), Punjab, India
parshotammehmi@yahoo.com

ABSTRACT

Security has always been a great concern for all software systems due to the increased incursion of the wireless devices in recent years. Generally software engineering processes tries to compel the security measures during the various design phases which results into an inefficient measure. So this calls for a new process of software engineering in which we would try to give a proper framework for integrating the security requirements with the SDLC, and in this requirement engineers must discover all the security requirements related to a particular system, so security requirement could be analyzed and simultaneously prioritized in one go. In this paper we will present a new technique for prioritizing these requirement based on the risk measurement techniques. The true security requirements should be easily identified as early as possible so that these could be systematically analyzed and then every architecture team can choose the most appropriate mechanism to implement them.

Keywords

Security Requirements, Threats, Vulnerabilities, Assets, Prioritization.

1. INTRODUCTION

Whenever there is the case of security issues attached with any particular system, it is of a great concern for the software applications. And so to develop a secure software application which would be efficient enough, there is always the need to use the security engineering process which consists of activities such as security requirement elicitations, analysis & prioritizations, specification and management, etc.In SDLC, [4] the security requirement process is generally defined with several functional and non functional requirements. And these days, the security of the software applications is the main priority so security requirements processes should be implemented during the different design and the development phases of the SDLC process of a particular system. And so we should also analyze and prioritize them according to the threat severity of a system.

DOI : 10.5121/ijcsea.2012.2303

And these days the applications for different softwares are also becoming very heterogeneous and vulnerable in some way. As reported in CERT [1], Security vulnerabilities have grown exponentially in this period. Attackers have sophisticated attack techniques to break security measures enforced by developers, which has lead to many crushing consequences like denial of services, etc.

Generally software engineering processes compels the security measures during the various design phases which results in using an inefficient measure. So this calls for a new software engineering process, so it is dedicated to designing, implementing and modifying software so that it is of high quality, affordable, maintainable, and fast to build. In simple terms the software engineering could be defined as the, "systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software, that is, the application of engineering to software"[2].

The basic idea in this research work will move around the concept of Security Requirements and its Prioritization. Prioritization is one of the important activities in the requirements engineering process, which aims in identifying the most fitting requirements for a specific release of a system. Generally, projects face limited resources such as short timelines, small budgets, restricted human power, and limited technology. As a result, projects often contain more candidate requirements that can be implemented in one product release time. Stakeholders need to decide which requirements should be implemented. [3]Requirements prioritization helps the project developers to select the final candidate requirements within their resource constraints. As Firesmith [4] has defined security requirement to be high level requirement that gives detailed specification of system behaviour that is not acceptable which is also distinguish these from security related architectural constraints. This is done so that requirement engineer can discover all the true security requirements. [5].

Before we can determine that a program is secure, we always have to determine that what exactly its actual security requirements. There is an international standard for identifying and defining these security requirements which are useful for many such circumstances which is the Common Criteria (CC).[6] The CC is basically used for the work to identify the information technology security requirements. There are also many other schemes which are available for defining security requirements and evaluating products to see if products meet the requirements, but some other schemes are generally focused on a specialized area and won't be considered further.

The requirements for eliciting different methods for a software are much, but we very less often see these elicitation performed specifically for security requirements. The one main reason for this is that few elicitation methods are specifically directed at security requirements. And another factor is that organizations not often address these security requirements elicitation specifically and instead chunk of them in with other traditional requirements elicitation methods. The requirements using templates [7] uses these different eliciting security methods but these are not integrated in conventional requirements engineering process.

2. RELATED WORK

In software engineering, the securities and its requirements must be discovered along with the other requirements of the system. Security requirements should be precise, adequate, complete and non- conflicting with other requirements. Once these requirements are clearly specified, they can then be implemented and maintained [4].

There is always a need to discover the requirement techniques which are presented by many other papers in their earlier work. The main concern would be the true security requirements identified as early as possible and systematically analyzed in such a way that we could present any technique for prioritizing these requirements based on risk measure techniques. So in this paper we will determine the relative necessity of the requirements, whereas all requirements are mandatory, some are more critical than others. So for this we need to propose a proper framework so these requirements could be prioritised efficiently.

As we are using different tools for this requirement prioritization framework we would use different methodologies such as STRIDE which classifies the schemes for characterizing the discovered threats according to the kinds of exploitation that are used [8], DREAD which provides a means to rate threats identified [9], CRAMM which simply calculate the measure of risk for each threat to an asset and vulnerability [10], etc. but there are some loopholes in these methods in terms of its implementation part or there process time. So we need to remove these loopholes in such a manner so that we could result into a framework which could combine these methodologies to prioritise the requirements subsequently removing all these loopholes.

And also there are many approaches are there for systematically performing this activity of requirement prioritization such as Numerical assignment which is a simple requirements prioritization technique based on grouping requirements into different priority groups. The number of priority groups can vary, but three is common i.e. “critical”, “standard”, and “optional”, or AHP in which it compares all possible pairs of hierarchical requirements to determine the priority, or Hundred Dollar Method and in this each stakeholder is asked to assume he/she has $100 to distribute to the requirements. The result is presented on a ratio scale. The ratio scale result can provide the information on how much one requirement is more/less important than another one. Most of these techniques are based on the attributes such as time, importance, cost, and the risk [3].

3. PROPOSED WORK

While Proposing a Security Requirement Prioritization Framework based on the threat analysis, we are giving a brief of all those steps which are necessary for achieving the final prioritized values of different security requirements and which are discussed below.

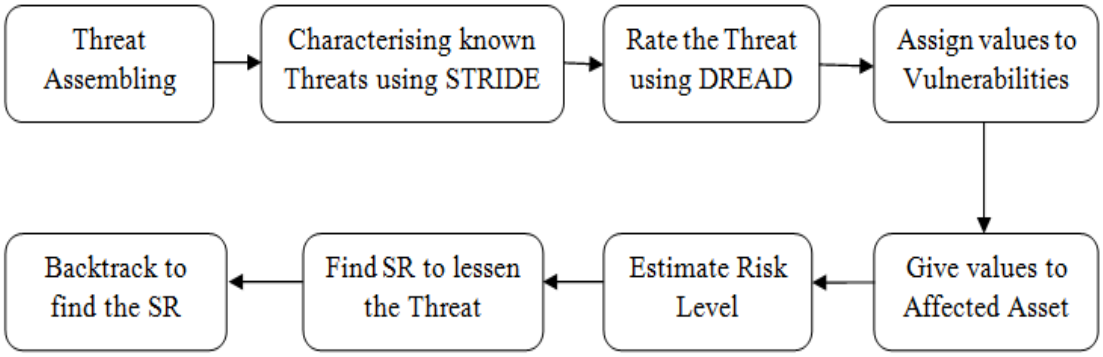

Figure 1. Proposed Framework for Security Requirement Prioritization

3.1 Assembling the Required Threats

In this we will assemble all those threats which are a source of each security requirements. As in common criteria based approach we shall be developing storage of deposits of all the threats[11]. Predefined threats can be retrieved from this storage area according to the profile of the user or the stakeholder and the list of all those predefined threats are:

a) Change Data
b) Data Theft
c) Deny Service
d) Disclose Data
e) Impersonate
f) Insider
g) Outsider
h) Privacy Violated
i) Repudiate Receive
j) Repudiate Send
k) Spoofing
l) Social Engineer

3.2 Characterising all the Known Threats using STRIDE Methodology

In this we generally classify the schemes for characterizing the discovered or known threats according to the kinds of exploitation that are used. It is used to simply help non-technical persons in the business world so that they could relate certain things according to their needs. This could be taken as a pen checklist which we use in our daily routine.[8]

3.3 Rate the Assembled Threats using DREAD Methodology

This method provides a means to rate threats identified and operates hand in hand with the STRIDE mechanism which categorizes threats. DREAD is an acronym, each letter of which stands for a threat attribute. Each of the attributes are ranked using one of 10 criticality ratings with being the lowest rating and 10 being the highest rating[9]. The attributes are :

a) Damage potential
b) Reproducibility
c) Exploitability
d) Affected Users
e) Discoverability

This algorithm is used to compute a risk value, which is an average of all five categories.
DREAD = (Damage + Reproducibility + Exploitability + Affected Users +Discoverability) / 5

3.4 Assigning the Values to Vulnerabilities

Vulnerability is defined as the weakness in the system that makes an attack more likely to succeed. All the values are generally project specific. Values of vulnerability are defined by the CRAMM method and these can be assigned according to following Table [10].

Table 1. Measure of Vulnerability

Condition	**Rating**
If an incident was to occur, there would be no more than a 33% chance of the worst case scenario.	Low (0.1)
If an incident was to occur, there would be a 33% to 66% chance of the worst case scenario.	Medium (0.5)
If an incident was to occur, there would be a higher than 66% chance of the worst case scenario.	High (1.0)

3.5 Values Given to Affected Assets

An asset can be anything that has a value to an organization (e.g. IT systems, information, staff, reputation, etc) and this is project specific. Different asset in a project is identified and their value is measured by weighing the impact of it when threat will occur.

3.6 The Risk Level is Calculated

Risk is defined [10] , as the probability that a threat agent (cause) will exploit system vulnerability (weakness) and thereby create an effect detrimental to the system. Here, the
Risk = Value based on Measure of Threat, Vulnerability and Asset.

After we have rated the threats, assigned vulnerability value and asset value we will use the 3 dimensional lookup table given by the CRAMM, in which we can calculate the measure of risk for each threat to an asset and vulnerability and then we can evaluate the threats based on their measure of risk.

3.7 Find the Security Requirement to Lessen the Threats

In this method we simply try to identify the various security requirements corresponding to threats so that we can give them priority to mitigate the threats.

3.8 Backtracking the Security Requirements Prioritizations

For this methodology we will identify the measures of risk to all the threats and prioritize them based on value of risk and assign them a final value for the prioritization.

4. CASE STUDY

Now in this case study, all the above discussed steps are defined below and are explained with an example of Air Reservation System. We have chosen various methods like STRIDE, DREAD, CRAMM method for final identification of risk as it offers structured and fast approach to risk analysis over other methods.

4.1 Threat Assembling

Using a common criteria based approach for assembling threats which are identified for our example are:

a) Change Data (CD)
b) Repudiate Receive (RR)
c) Spoofing (Sp)
d) Flooding (F)
e) Privacy Violated (PV)
f) Outsider (O)
g) Physical (P)

4.2 Characterising Known Threats

While threats are characterised, this uses the STRIDE model which is an acronym for six threat categories that are listed below :

a) Spoofing identity (S)
b) Tampering with data (T)
c) Repudiation (R)
d) Information disclosure (I)
e) Denial of service (D)
f) Elevation of privilege (E)

The tabular representation is shown in the table 2, where the assembled threats are correspondingly shown with their categories of these threats using the number 1 at their identified place.

Table 2. Checklist for Discovered Threats

	S	**T**	**R**	**I**	**D**	**E**
CD	-	1	-	-	-	-
RR	-	-	1	-	-	-
Sp	1	-	-	-	-	-
F	-	-	-	-	1	-
PV	1	-	-	1	-	-
O	-	-	-	-	-	1
P	-	-	-	-	-	1

4.3 Rating Threats

For the calculation of the overall risk for different threats, it always produces a number between 0 and 10, the higher the number, the more serious the risk. And these could be then scaled up or down according to the needs. As in our case we are scaling down these values because of the CRAMM matrix used for further calculations in the next forthcoming steps, this could be seen in the table 3.

Table 3. Measure of Risk

ID	D	R	E	A	D	Overall Risk
CD	5	5	5	10	5	6
RR	10	5	5	10	5	7
Sp	10	10	5	10	5	8
F	0	10	5	0	5	4
PV	5	5	5	0	0	2
O	10	0	5	10	10	7
P	0	0	5	0	5	2

Now, after this risk rating method, we can simply scale down our observed 10-point scaled values lower to the 2-point scaling system as shown in table 4.

Table 4. Scaling down the Values

DREAD 10 point Scale	DREAD 2 point Scale
1-2	Very Low
3-4	Low
5-6	Medium
7-8	High
9-10	Very High

Whatever threats have been identified we have to assign them a value according to CRAMM evaluation so for this we have to consider the table 5 and here all values are project specific and are taken by observation.

Table 5. Measure of various Threats

Threats with overall risk	Level of Threat	Value (.1,.34,1,3.33,10)
CD (6)	Medium	1
RR (7)	High	3.33
Sp (8)	High	3.33
F (4)	Low	0.34
PV (2)	Very low	0.1
O (7)	High	3.33
P (2)	Very low	0.1

4.4 Assigning Value to Corresponding Vulnerability

Here the values of vulnerabilities are defined by CRAMM method and will be taken as low (0.1), medium (0.5) and high (1).

4.5 Give Values to Affected Assets

The different assets identified in our particular example are :

a) Traveler Information
b) User Login Information
c) Credit Card Information
d) Communication Channels
e) Ticket Information

Now we have to define various vulnerable assets that will be affected that corresponds to the threat are as in Table 6.

Table 6. Possible Vulnerable Assets

Threat	Affected Assets
CD	Traveller information, Ticket Information
RR	User Login Information, Credit Card Information
Sp	Credit Card Information, Communication Channel
F	Credit Card Information, Traveller Information
PV	Ticket Information
O	Communication Channel, User Login Information
P	User Login Information

And the different values of various assets are depicted in Table 7.

Table 7. Measure of Assets

Asset	Value (1 to 10)
Traveler information	7
Ticket Information	5
Credit Card Information	9
User Login Information	4
Communication Channel	6

4.6 Estimate the Risk Level

After we have rated the threats, assigned vulnerability value and asset value we will use the 3 dimensional lookup table given by the CRAMM where the strength of the threat, the level of the vulnerability and the value of the asset are input parameters, gives the final value of risk in the range 1 through 7.

For eg. suppose asset is Credit Card Information (9) the threat to this is Spoofing (3.33) and Vulnerability being medium (0.5) the measure of risk will be 6. In the similar fashion we can calculate the measure of risk for each threat to an asset and vulnerability and then we can evaluate the threats based on their measure of risk as given in the Table 8.

4.7 Identify the Security Requirements

Here the different security requirements corresponding to the threats in the table are identified. For eg. the measure of the risk for the change_data for affected assets ticket and traveller info is 4 and 5 respectively, so its threat prioritization value will be the sum of them i.e. 9. And the total measure of risk for the repudiate_receive for affected assets like user login info and credit card info will be 4 and 6 respectively and which will be added to form its threat prioritization value to total of 10 as given in the table 8.

4.8 Backtrack to Find Priority of Security Requirement

This particular step is the real part of our progress what we have done till now as in this we will backtrack all the gathered values of different threats which are discovered earlier and assign them a final value which will prioritize all the security requirements. For.eg. the value of threat prioritization for change_data is 9 and for repudiate_receive is 10, so here the overall security requirement prioritization value say authorization will be a sum of them which is 19.

Table 8: Detailed computed values of Security Requirements

Security Requirement	Threats Discovered	Threat Rating	Vulner- - ability	Affected Assets	Asset Value	Risk Estimate	Threats Prioritization	Security Requirement Prioritization
Authentication	1.Spoofing 2.Privacy_Violated 3.Outsider	3.33 0.1 3.33	0.5 0.1 0.1	Credit Card Info(1) Comm. Channel(1,3) Ticket Info(2) User Login Info(3)	9 6 5 4	6 5,4 2 3	11 2 7	20
Authorization	1.Change_Data 2.Repudiate_Receiv	1 3.33	0.5 0.5	Ticket Info(1) Traveller Info(1) User Login Info(2) Credit Card Info(2)	5 7 4 9	4 5 4 6	9 10	19
Integrity	1.Flooding	0.34	1	Traveller Info(1) Credit Card Info(1)	7 9	6 5	11	11
Identification	1.Outsider	3.33	0.1	Comm. Channel(1) User Login Info(1)	6 4	5 3	8	8
Privacy	1.Privacy_Violated	0.1	0.1	Ticket Info(1)	5	2	2	2

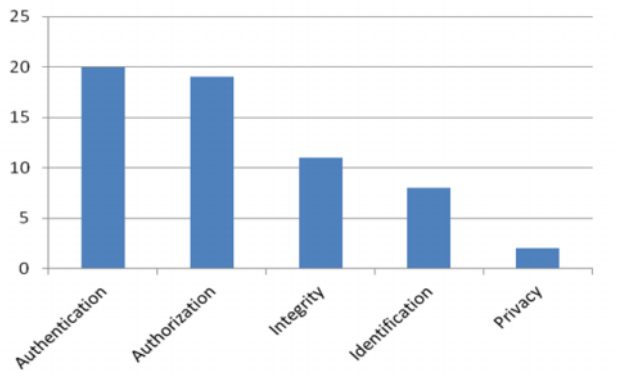

Figure 2. Measure of Security Requirement Prioritization

5. CONCLUSION

In this paper we have reviewed the security engineering process for eliciting security requirements and process these to prioritize based on risk estimation techniques, and then will identify the threats, analyze them and finally evaluate them so that all the threats that may occur in the project later will be detected in early phase. At last the associated risk will be estimated that will help in security requirement prioritization and so a good quality secure software is developed. As in our case study, we have concluded the values of Air Reservation System as shown above in figure 2.

6. ACKNOWLEDGEMENT

Our thanks to the experts who have contributed towards the development of the template.

7. REFERENCES

[1] Software engineering institute website, www.cert.org
[2] Roger S. Pressman (Fifth Edition), Software Engineering-A Practitioner's Approach, McGraw Hill, p.20- 24.
[3] Qiao Ma, (2009) " The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements"
[4] Donald G. Firesmith, (2003) "Engineering Security Requirements", Journal of object technology, vol 2, no.1, pp.53-68.
[5] Agarwal A, Gupta D, (2008) "Security Requirement Elicitation Using View Points for online System", IEEE Computer Society.
[6] Security Requirements, http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/requirements.html
[7] Sindre G, Opdahl AL, (2001) "Templates for Misuse Description". Proceeding 7th Int'l Workshop Requirements Eng.: Foundation for Software Quality.
[8] Dle R. Thomson, Neeraj Chaudhary, (2006) "RFID Security Threat Model", IEEE papers.
[9] Supreet Venkataraman, (2007) "Prioritization of Threats Using the k/m Algebra", IEEE papers.
[10] CRAMM's Assessment of Measure of Risk, http://www.cramm.com/files/techpapers/CRAMM%20Countermeasure%20Determination%20and%20Calculation.pdf
[11] Michael S. Ware,(2006) "Using the Common Criteria to Elicit Security Requirements with Use Cases", IEEE papers.

Authors

Aayush Gulati is a student of Lovely Professional University pursuing Masters of Technology in Computer Science and Engineering. Shalini Sharma is an Assistant Professor in Lovely Professional University and has done her Masters of Technology in Computer Science and Technology from Delhi Technical University. Parshotam Mehmi is a student of Lovely Professional University pursuing Masters of Technology in Computer Science and Engineering.