journal homepage: www.sabauni.net/ojs

Saba Journal of Information Technology and Networking (SJITN)

Article

Measure Security Requirement of Critical System Using Fuzzy Logic

Dr. Hazaa M. *, Gassem,Y. , Alzidani, M. , Alansi, S. , Albadani, A.

Thamar University, Yemen

Article info

Article history:

Accepted Dec, 2016

Keywords:

Security requirement specification ,
Fuzzy logic ,
Critical system ,
Threat

Abstract

Security is one of the important dependable system dimensions, as it plays a primary role in a critical system that must be dependable and provides acceptable degrees of security, safety and integrity. To achieve success in this type of system development, and to avoid loss and failure of the system that we want to develop, it is important to be sure that the system specification that relates to the security is correct and can be developed in organization's abilities and that the required security level for the requirements specification of the critical system is acceptable to the organization, and can be developed in terms of costs, technology and assets. For this purpose, we will build a method that will perform the analysis, support and confidence ensuring that these requirements are acceptable and whether it can be applied in the system or not. Our method will use fuzzy logic to capture knowledge from analysis experts as rules that would help take a certain decision with respect to asset values, available technology and threats of the organization. From this, our methodology will give the analyst a level of confidence and acceptance for the security Requirement specification that we deal with.

* Corresponding author: Dr.Muneer Hazaa
E-mail: muneer_hazaa@yahoo.com

1. Introduction

Any critical system must always be secure. The systems that are not secure will not be dependable, and consequently will affect other dependability factors like availability, reliability and safety [1]. So, it is important to build a plan and test the software process specification's requirements before we build such a type of critical system that can be applied, and to be sure to some degree of certainty that the requirements are acceptable and applicable [16] .

In fact, the security depends on many factors which in most cases cannot be covered because of the spread of security domain. Also, types of attacks and technology that can be implemented in this kind of system are variable. Furthermore, there are several processes for identifying and prioritizing risks. One of the most effective processes is threat modeling. Threat modeling is the process of identifying, quantifying and analyzing potential threats of a computer-based system. It is a process of assessing and documenting a system's security risks [4]. It involves identifying the key assets of an application, decomposing the application, identifying and categorizing the threats to each assets or component, rating the threats based on a risk ranking, and then developing threat mitigation strategies that are then implemented in design, code and test cases [5]. Categorizing threats is the first step toward effective mitigation [4].Threats can be classified into six classes based on their effect [6]. This is generally referred to as the STRIDE model. The STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privilege) model was used by Microsoft for categorizing threats [5]. Security measurements can be achieved by estimating the ability of building these requirements that we deal with in the specification phase. This method will be useful to support the Analyst who doesn't have enough experience with such kind of system and help support to achieve the optimal level of certainty regarding these requirements.

2. Security Requirements with fuzzy logic

When security requirements are considered at requirements specification stage writing from the system development life cycle, they tend to be general lists of security features, such as password protection, firewalls, virus detection tools, and the like [10]. These are, in fact, not security requirements at all but rather implementation mechanisms that are intended to satisfy unstated requirements, such as authenticated access. As a result, security requirements that are specific to the system and that provide protection of essential services and assets are often neglected. In reviewing requirements documents, we typically find that security requirements, when they exist, are in a section by themselves and have been copied from a generic set of security requirements. The requirements elicitation and analysis, needed to get a better set of security requirements, seldom take place and measure these requirements if it will achieve the level of security required [11]. A proposed model consists of three steps that generate a numerical value in which the degree of security available appears and helps to detect prioritized security requirements and accept or reject this system depending on measurement value.

3. Literature Review

Security is a system attribute that reflects the ability of the system to protect itself from external attacks that may be accidental or deliberate [1]. The specialized terminologies associated with security are exposure, vulnerability, attack, threats and control [2].The assessment of system security is becoming increasingly important as more and more critical systems are Internet-enabled and can be accessed by anyone with a network connection [1]. These types of security assessment are very difficult to carry out. Consequently, systems are often deployed with security loopholes that attackers use to gain access to or damage these systems [3]. It is very difficult for end-users of a system to verify its security. Consequently, bodies in North America and Europe have established sets of security evaluation criteria that can be checked by specialized evaluators. Software product suppliers can submit their products for evaluation and certification against these criteria [7]. Therefore, if you have a requirement for a particular level of security, you can choose

a product that has been validated to that level [1]. However, many products are not security-certified or their certificate applies only to individual products. When the certified system is used in conjunction with other uncertified systems, such as locally developed software, the security level of the overall system cannot be assessed [1].

Fuzzy Logic introduced by Zadeh (1965) gives us a language, with syntax and local semantics, in which we can translate our qualitative knowledge about the problem to be solved [8]. Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing [14]. Fuzzy logic provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. There are many factors which account for the increase in question but the most prominent among them is the rapidly growing use of soft computing and especially fuzzy logic in the conception and design of intelligent systems. As one of the principal constituents of soft computing, fuzzy logic is playing a key role in the conception and design of various systems [15]. There are two concepts within fuzzy logic which play a central role in its applications. The first is that of a linguistic variable, i.e., a variable whose values are words or sentences in a natural or synthetic language. The other is that of a fuzzy if-then rule in which the antecedent and consequent are propositions containing linguistic variables [8]. The essential function served by linguistic variables is that of granulation of variables and their dependencies. In effect, the use of linguistic variables and fuzzy if-then rules results -through granulation in soft data compression which exploits the tolerance for imprecision and uncertainty.

In fact, the theory of fuzzy sets theory is a generalization and extension of conventional nature which agrees with the language and understanding of human nature as well [13].

Definition 1.Suppose that X is a set of reference, the common characteristic of a subset A of X, is defined as follows:

$$\mu_A(x) = \begin{cases} 1 & : \ x \in A \\ 0 & : \ x \notin A \end{cases} \qquad (1)$$

According to the above definition, for each X Ɛ x, it will be only one of the values 0 or 1.

Definition 2. If the range of the function μA of the [1, 0] to the interval [1, 0] expands, we have a function to every member of X, the number in the range [1, 0] assigns. The other set A is not normal, but is called a fuzzy set (A is a fuzzy subset of X). In the above definition, if μA (X) Ɛ [0,1] then the membership of x belongs to variable A, with a certain degree between [0,1]. In fact, here is an extended concept of membership of an element. It also represents the membership degree of μA (X), the membership in a fuzzy set is the element x. If the degree of membership of an element is set to zero, the member is fully withdrawn [9]. And if the degree of membership of a member is set to one, the member is quite a collection. If the degree of membership of a member is between zero and one, the number that indicates the degree of membership is gradual. Figure 1 is an example of the membership function of a fuzzy set.

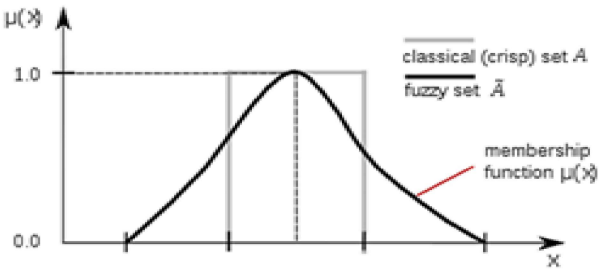

Fig. 1.Membership Function of a Fuzzy Set

In this paper we use a Gaussian membership function. A Gaussian membership function can be demonstrated by the following equation:

$$\mu_{A^i}(x) = \exp(-\frac{(c_i - x)^2}{2\sigma_i^2})$$

3.1 Fuzzy logic operation steps are described as follows:

Step1.Fuzzy Inputs:

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In Fuzzy Logic Toolbox™ software, the input is

always a crisp numerical value limited to the universe of discourse of the input variable (in this case the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation [12].

Step2.Apply Fuzzy Operator:

After the inputs are fuzzified, you know the degree to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value [12].

Step3.Apply Implication Method:

Before applying the implication method, you must determine the rule's weight. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Generally, this weight is 1 (as it is for this example) and thus has no effect at all on the implication process. From time to time you may want to weight one rule relative to the others by changing its weight value to something other than 1. After proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which appropriately weights the linguistic characteristics that are attributed to it. This consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule. Two built-in methods are supported, and they are the same functions that are used by the AND method: min (minimum), which truncates the output fuzzy set, and prod (product), which scales the output fuzzy set. [12].

Step4.Aggregate All Outputs:

Because decisions are based on the testing of each rule in an FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable [12].

Step5.Defuzzify:

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set [12].

4. Design Methodology

In this section, we will explain the proposed methodology to determine the possibility of applying security requirements according to the specification criteria which is related to the security requirements in the critical systems, where the criteria which we use in the methodology is Assets in the system and the possible or expectant threats on this assets. The whole task starts with identifying the scope of the product. Asset based risk management is then conducted to identify the risks for all critical assets. Critical assets are identified based on costs of production and reproduction, the amount of loss for any damages, etc. Possible threats and vulnerabilities to this critical asset are then determined through threat profiles, attack trees, threat sources etc. Identification of asset, threat and vulnerability related to these assets are critical elements for risk identification. These risks are then analyzed by the likelihood of occurrence and by estimating their negative impact. The negative impact can be computed according to proportions between 1 and 100. If this proportion is low, that means that it has a low risk impact, and if the result is of a high value, then the result will be a high risk impact. Finally,

a mitigation plan, protection strategies and action lists are suggested to control the risk at an acceptable level. Security goals and policies are then outlined considering the product and organization. Security goals are the organization's motivation and business gain by applicability of the management control principles. Security policy sets out conditions to achieve the security goals

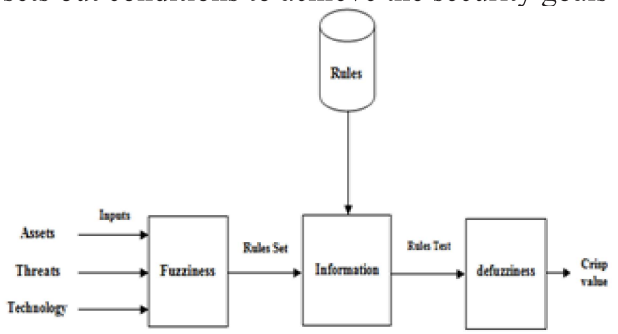

Figure 2: Proposed Method Architecture for Fuzzy Logic-based Assets, Threat and Technology Modeling

The steps involved in the design are:

First. Fuzzification Phase:

- Determining the cost of the assets (Memory, Operating System, Internet Explorer, Wired Cable, Sensor, Staff and Web Site) that are related to software and system. Here, we mean by assets, data and application. The value of the assets is determined according the system analysts and their experience or knowledge on the assets. Assets will determine the level of costing which may be cheap, normal, expensive or very expensive.

2. Linguistic variable: Assets

Table 1. The input linguistic variable (Assets)

Linguistic value	Numerical range
Cheap	[-0.3333 0 0.3333]
Normal	[0 0.3333 0.6667]
Expensive	[0.3333 0.6667 1]
Very-Expensive	[0.6667 1 1.333]

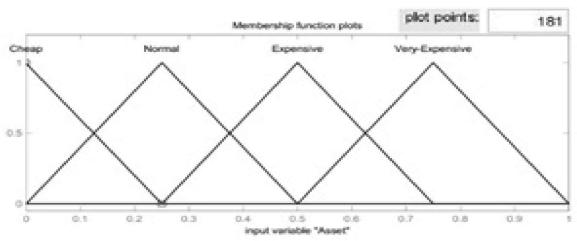

Fig. 3: Assets Value on Degree of Membership

- Determining the possible threats (Attacker , Resist, Damage , Risk , Unwanted properties , Constraint , Follow register , Exponent , Exposure , Denial of service , Mistakes ,Omission ,Vulnerability , maintenance ,install , download, configuration ,updating , number version , reparability, response , design , number activities within program, monitor attack , determining , survivability , reparability, environments) on this asset and determining the level of hazardousness. Also, the system analyst determines the level of hazardousness of threats for this requirement which may be: Intolerable , as low as practical , acceptable.

Table 2. The input linguistic variable (Threats)

Linguistic value	Numerical range
Low	[-0.4 0 0.4]
Medium	[0.1 0.5 0.9]
High	[0.6 1 1.4]

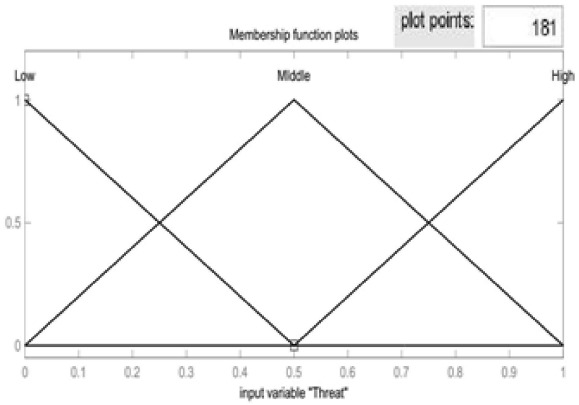

Fig. 4: Threats Value on Degree of Membership

-Determining the existing techniques (Antivirus, protocols, validation tools, verification tools, test tools, authentication, session use, encryption tools, documentation, password, applications and Operating System) in the markets and their costs.

Table 3. The input linguistic variable (Technology)

Linguistic value	Numerical range
Easy-to-get	[-50 0.3 0.7]
Hard-to-get	[0.25 0.7 50]

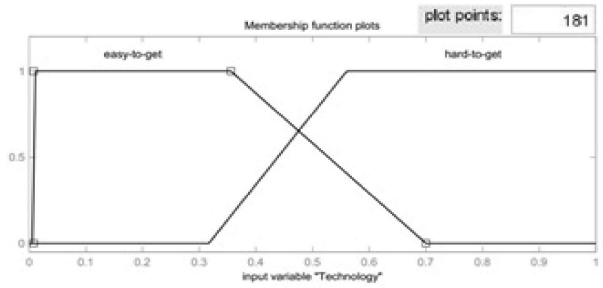

Figure 5: Technology Value on Degree of Membership

- These steps (1,2,3) are the inputting phase for the methodology under the name Fuzzification, where they will transform form the verbal variable for the inputting of assets, threats and technology into variables for the methodology or following

Second: Rule and Inference Phase:

In the Phase of Rule and inference, the output of the Fuzzification phase is applied to the input linguistic variable assets. Technology and threats apply the rules on these variables. After that the output of each linguistic variables is applied on mamdani inference function to get the result of the risk.. Based on these results we can determine the level of possibility of applying this requirement or not.

Table 4. Samples of rules

Rules
1. If (Technology is Easy-to-get) and (Threats is Low) and (Assets is Very-Expensive) then (Level-of-Requirement-Risk is Acceptable)
2. If (Technology is Easy-to-get) and (Threats is Low) and (Assets is Normal) then (Level-of-Requirement-Risk is Acceptable)
3. If (Technology is Easy-to-get) and (Threats is Low) and (Assets is Expensive) then (Level-of-Requirement-Risk is ALARP) (1)
4. If (Technology is Easy-to-get) and (Threats is Low) and (Assets is Very-Expensive) then (Level-of-Requirement-Risk is ALARP)
5. If (Technology is Easy-to-get) and (Threats is Medium) and (Assets is Cheap) then (Level-of-Requirement-Risk is Acceptable) (1)
6. If (Technology is Easy-to-get) and (Threats is Medium) and (Assets is Normal) then (Level-of-Requirement-Risk is ALARP) (1)
7. If (Technology is Easy-to-get) and (Threats is Medium) and (Assets is Expensive) then (Level-of-Requirement-Risk is ALARP) (1)
8. If (Technology is Easy-to-get) and (Threats is High) and (Assets is Cheap) then (Level-of-Requirement-Risk is ALARP) (1)
9. If (Technology is Easy-to-get) and (Threats is High) and (Assets is Normal) then (Level-of-Requirement-Risk is ALARP) (1)
10. If (Technology is Easy-to-get) and (Threats is High) and (Assets is Expensive) then (Level-of-Requirement-Risk is Unacceptable) (1)
11. If (Technology is Hard-to-get) and (Threats is Low) and (Assets is Cheap) then (Level-of-Requirement-Risk is ALARP) (1)
12. If (Technology is Hard-to-get) and (Threats is Low) and (Assets is Normal) then (Level-of-Requirement-Risk is ALARP) (1)
13. If (Technology is Hard-to-get) and (Threats is Low) and (Assets is Expensive) then (Level-of-Requirement-Risk is Unacceptable) (1)
14. If (Technology is Hard-to-get) and (Threats is Medium) and (Assets is Expensive) then (Level-of-Requirement-Risk is Unacceptable) (1)
15. If (Technology is Hard-to-get) and (Threats is High) and (Assets is Expensive) then (Level-of-Requirement-Risk is Unacceptable) (1)

Third: Defuzzification Phase:

After the inference rule phase is finished, which has been handled based on the input related to the security requirement in Fuzzification phase, the Defuzzification will transform and output the result of inference to a specific range that represents the risk of developing this requirement and find out whether the requirement is accepted or not according to the criteria mentioned above.

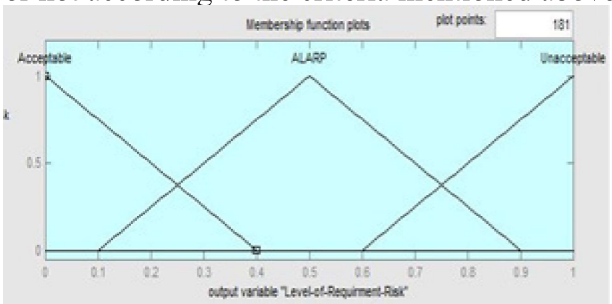

Fig.6: Risk Value on Degree of Membership

Table 5. The output linguistic variable

Linguistic value	Numerical range
Cheap	[-0.3333 0 0.3333]
Normal	[0 0.3333 0.6667]
Expensive	[0.3333 0.6667 1]
Very-Expensive	[0.6667 1 1.333]

5. Implementation and Evaluation

The methodology is implemented using MATLAB fuzzy logic toolbox. Implementations are presented below:

1- FIS Editor (Figure 6):

This window is used to select a new FIS type with any particular model, we can add the related variables to the mode, and input or output variable names can be added as well. In our methodology, we chose the model of Mamdani.

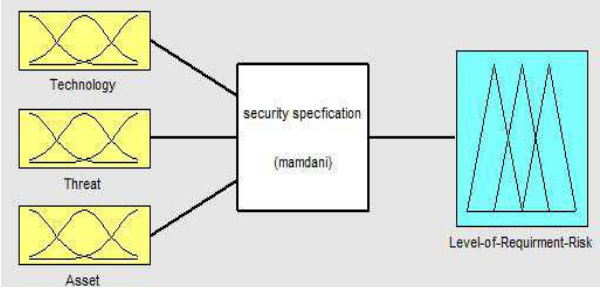

Fig. 7: FIS Editor

2- Membership function editor (Fig. 3 and 4):

This window is used for the input or the output of the membership function that can be added or removed. It also makes it possible to specify the ranges of each of the variables and membership functions.

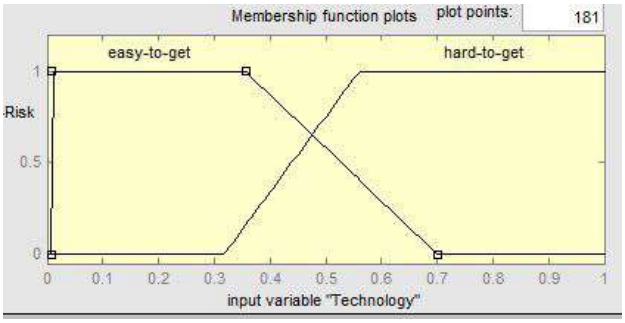

Fig. 8: Technology Input Variable

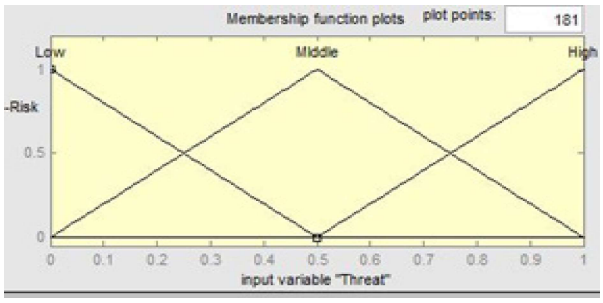

Fig. 9: Threat Input Variable

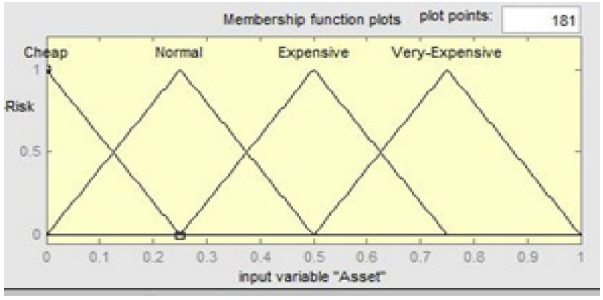

Fig. 10: Asset Input Variable

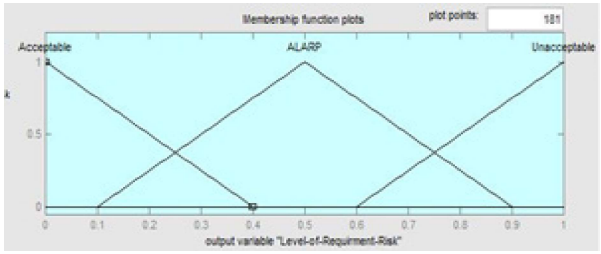

Fig. 11: Risk Level Output Variable

3- Rule editor (Figure 12):

This is used to add, change or delete rules. It provides opportunity to change the connections and weight applied to the rules (the default weight is always 1).

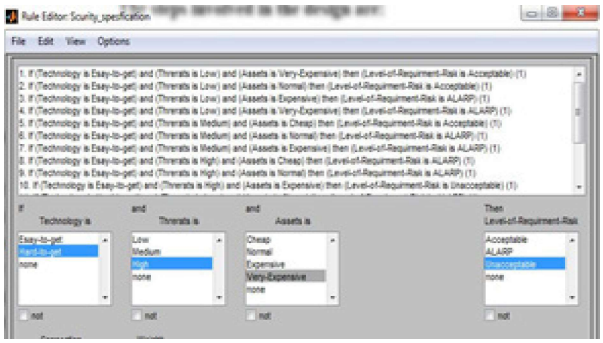

Fig. 12: Rule Editor

4- Rule Viewer (Figure 13):

The rule viewer shows a graphical representation of each of the variables through all the rules, a representation of the combination of the rules, and a representation of the output from the defuzzification. It also shows the crisp value output of the system. Data are entered for analysis through the Rule Viewer at the Input text field.

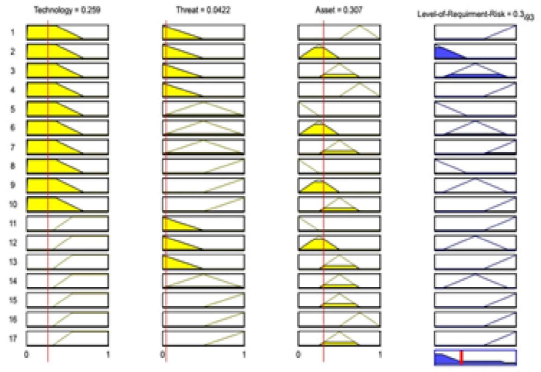

Fig. 13: Rule Viewer

As shown the method gives us the level of risk in the previous figure. The next figure, 13 shows the relation and the effect of assets with threat on the level of risk

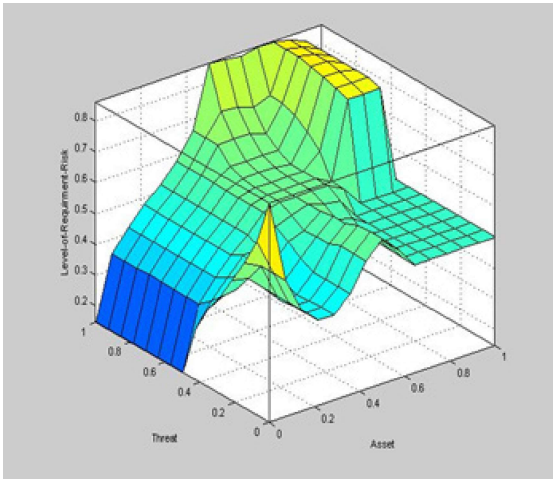

Fig. 14: Asset & Threat

The relation between asset and technology is also shown in figure 14

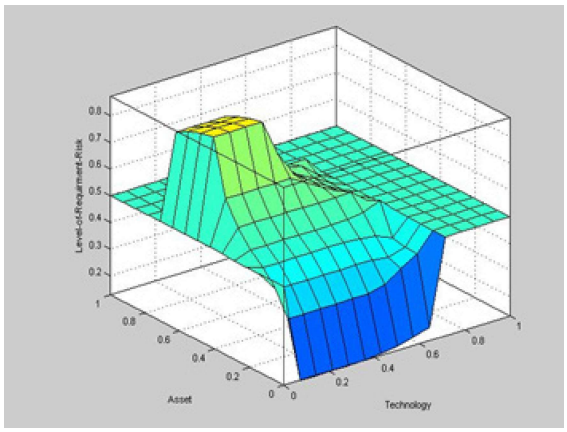

Fig. 15: Asset &Technology

6. Evaluation

In this section we will test the methodology and find out the results of experiment. The parameters of the input can be adapted by moving the value of any input variable in the rule viewer.

First Scenario:

Fig. 16: Second Scenario Result

As we see in Figure 16. When the Threat of the security requirement was high with 0.922 %, assets of the organization were very expensive with 0.861 % and Technology was hard to get with 0.633 %. The model predicted that requirement have higher risks with 0.853 %. So the analysis should consider this requirement and find a suitable solution

Second Scenario:

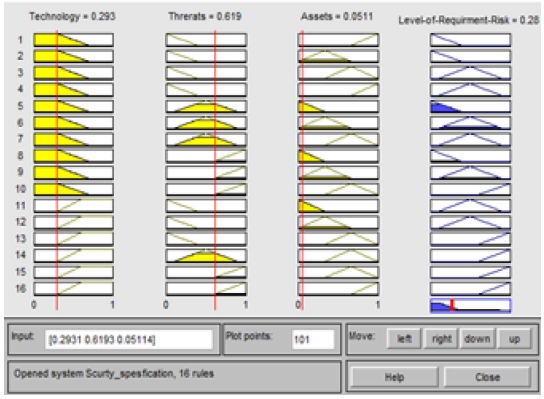

Fig. 17: Second Scenario Result

As we see in Figure 17, when the threat of security requirement was medium with 0.619 %, assets of the organization were very cheap with 0.0511 % and technology was easy to get with 0.293 %. The model predicted that the requirement has a higher risk with 0.28 %. So the risk of this requirement can be acceptable and the analysis can devolve it with a confidence of 78%.

Third Scenario:

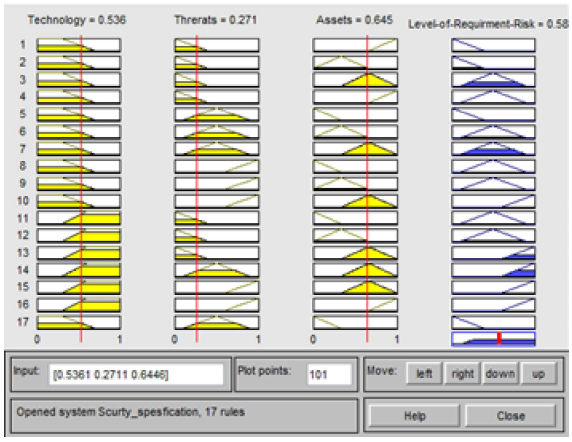

Fig. 18: Third Scenario Result

As shown in Figure 18, when the threat of the security requirement was Low with 0.271 %, assets of the organization were very expensive with 0.645 % and technology was easy to get with 0.536 %. The model predicted that the requirement has a higher risk with 0.58 %. So the risk of this requirement can be acceptable so the analysis can devolve it with a confidence of 42%.

As a result, these tools can be helpful for project team analysis that can support and describe the degree of any requirement after adding all rules to the database of those rules.

7. Conclusion

In this work, a fuzzy based system was designed to evaluate the possibility of applying security requirement throw specification phase, because it is impossible to provide assurance for the system and justify security measures incorporated unless the system is analyzed during the designing state of computer based systems. With this system designed, risk analysis has been made easier to estimate.

8. Future work

For further research, this system is enhanced by redesigning the methodology to dedicate the value of risk for given requirements and then suppose appropriate solutions to mitigate the possible risk.

References

[1] Sommerville, I.(2007). Software Engineering .BCS/IEE Software Eng. J., 11 (1), 5–18.

[2] Pfleeger, C. P. (1997). Security in Computing, 2nd edn. Englewood Cliffs, NJ: Prentice Hall.

[3] Anderson, R. (2001). Security Engineering: A Guide to Building Dependable Distributed Systems. Chichester: John Wiley & Sons

[4] Ambler, W. S (2005). Introduction to security threat modeling. Agile Modeling. Available at http://www.agilemodeling.com/artifacts/securityThreatModel.htm

[5] Davis, N., Howard, M., Humphrey, W., McGraw, G., Redwine Jr., S. T., Zibulski, G., & Graettinger, C. (2004). Processes to produce secure software: Towards more secure software. A report at National Cyber Security Summit, Vol. 1. Available at http://www.cigital.com/papers/download/secure_software_process.pdf

[6] Sodiya, A.S, Onashoga, S.A, &Ajayi, O.B (2006). Toward Building Secure Software Products. Journal of issues in Informing Science and Information Technology, 3,635-646. Available at http://informingscience.org/proceedings/

InSITE2006/IISITSodi143.pdf Swiderski, F. & Snyder, W. (2004). Threat modeling. Microsoft Press Professional Book Series

[7] Gollmann, D. (1999). Computer Security. Chichester: John Wiley & Sons.

[8] Zadeh L.A., Fu K.-S., Tanaka K. (2014), Fuzzy sets and their applications to cognitive and decision pro¬cesses.Proceedings of the US–Japan Seminar on Fuzzy Sets and TheirApplications. Academic Press University of California, Berkeley, Califor¬nia, July 1-4.

[9] Ross T.J. (2013), Fuzzy logic with engineering applications. Wiley, Vol. 761.

[10] Toval, Ambrosio, et al. "Requirements reuse for improving information systems security: a practitioner's approach." Requirements Engineering 6.4 (2002): 205-219.

[11] Mead, Nancy R., and Ted Stehney.(2005) Security quality requirements engineering (SQUARE) methodology. Vol. 30. No. 4. ACM.

[12] Zimmermann, H-J.(1996) "Fuzzy Control." Fuzzy Set Theory—and Its Applications. Springer Netherlands, 203-240.

[13] Dubois, Didier, and Henri Prade (2012), eds. Fundamentals of fuzzy sets. Vol. 7. Springer Science & Business Media.

[14] Ali, Mohd Hasan, Toshiaki Murata, and Junji Tamura (2004). "The effect of temperature rise of the fuzzy logic-controlled braking resistors on transient stability." IEEE Transactions on Power Systems 19.2 : 1085-1095.

[15] Zadeh, Lotfi A. (1998) "Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems." Soft Computing-A fusion of foundations, methodologies and applications 2.1 : 23-25.

[16] Reena Dadhich B.M.(2012), Measuring reliability of an aspect oriented software using fuzzy logic approach. International Journal of Engineering and Advanced Technology (IJEAT).