Asian Journal of Information Technology 15 (11): 1758-1775, 2016
ISSN: 1682-3915

Software Security Risk Assessment of Data Communication Network Through Attack Decomposition Using Fuzzy Rough Sets

[1]D. Kavitha, [2]S. Chandrasekaran and [3]M. Vigilson Prem
[1]Department of Computer Science and Engineering, Valliammai Engineering College, 603203 Chennai, India
[2]Department of Computer Science and Engineering, Sri Ranganathar Institute of Engineering and Technology, Coimbatore 641110, India
[3]Department of Computer Science and Engineering, R.M.D Engineering College, Tami Nadu, India

Abstract: The objective of the work is to propose a software risk analysis model considering a communication infrastructure with various types of attacks in network and information security processes using fuzzy rough set based approach. The information and network security mechanism are applied as individual services and these services are provided by various vendors based on the data transactions across multiple physical and virtual computing networks. A systematic and scientific approach using mathematical modelling of the events and activities are needed in order to prevent the application, service and data in the security domain. In the case of health care, banking transactions and vehicular tracing application with numbers of devices and computing nodes, a risk analysis is needed before attempting to plan and implement any security decision making solutions. The respective rules and regulations of each subsystem and the depth of each and every problem components are to be considered to model a security plan and operations running with many uncertain events in a communication infrastructure. The compatibility of all the complex events and the compliance of individual requirements are to be considered quantitatively. The earlier STRIDE and OCTAVE security risk models focus on security goals, activities in the organization not focusing on the various root causes of those risks. In business and statistical computing, the different forms of noncompliance, causality and composability of multiple software components are to be considered in determining the process risks but also lead to complexity and controllability of the vulnerable features to minimize the risks due to people during the evolution of any product. In the context of software security, the risk analysis has been carried out in the design stages focusing on the network, hardware software, hacker related features to accept and then assess the risk as per the proposed risk acceptance and risk assessment functions respectively. The proposed SIX CAUSE model exploes the various causes for the risks due to the vulnerability of any web or cloud application or a service with all its compliances before putting into operational modes. The risk pyramid and the minimum rules that are needed to analyse the various types of risks with the help of a decomposition tree and the sub components in the security area are studied to determine the application or system software security risk as Risk Assessment Document (RAD) using fuzzy rough set theory.

Key words: Software attacks, risk analysis, communication infrastructure, risk assessment, risk acceptance, risk pyramid, fuzzy rough set

INTRODUCTION

Software compliance management and control over the enforcement of regulatory compliances is the need of the networked world since the security risk level due to a service mismatch is higher than many functional faults (Hamdaqa and Hamou, 2011). The faults due to incorrect timing of control signals or some flaws in the security features may not affect the system performance in the long run of the software application or service within a communication infrastructure. At the same time a small deviation from the process requirements and training will make the system incompatible with the existing standards which are the root causes for many types of computer communication failures. The security risks due to these factors can range from people to hardware and software to system deployment. The hardware flaws and software vulnerabilities invite a special kind of risks associated

Corresponding Author: D. Kavitha, Department of Computer Science and Engineering, Valliammai Engineering College, 603203 Chennai, Tamil Nadu, India

with the security of the underlying system including the network and information. These types of security risks are due to software models, design, code and deployment phases of the software. The software risk analysis for security risk is considered as significant and indispensable process in all phases of software development life cycle (McManus, 2004). Security risk can be a state of holding some or zero variation or likelihood of loss in the potential level of a set of expectation. This set of expectations influenced by the income or outcome of many chosen actions and inactions these actions can be performed partially or fully in a given context either willingly or unwillingly. Security risk occurs due to noncompliance whereas noncompliance is defined as the failure or refusal to conform to certain rules, regulations and standards (HajSaid *et al*., 2011). The security risk can be defined with two parts, the probability of security failure and the consequence of security failure and its severity. The probabilities of security failure events due to noncompliance of each software element are estimated first and the severity of ecurity failures of those elements is estimated based on fuzzy rough set properties (Jensen and Shen, 2007). The different forms of noncompliance are mainly of the types, procedural noncompliance, regulatory noncompliance, people noncompliance and quality noncompliance as per the respective standards of the application software development. The software security risk, compliance has to be compared with the standard regulation and legal compliance before the corresponding services connected to the customer (Ni *et al*., 2003). The level of noncompliance and the amount of noncompliance and standard of events and methods are mostly unknown and imperfect. Hence, a rough set theory is used as a new mathematical tool to deal with this incomplete, imprecise and uncertain knowledge (Jensen and Shen, 2004). In this theory, the rough set membership function expresses the conditional probability that an object belongs to a set given the relation and can be interpreted as a degree that the object belongs to the set in view of information about the object expressed by the relation (Yeung *et al*., 2005).

The software security risk analysis consists of five phase’s intelligent discovery, automated risk detection, testing and validation, actionable reporting, remediation and mitigation. The important phase in security analysis is the initial intelligent discovery which detects threats and so, it is necessary to measure the likelihood of occurrences of the flaw in the security events (Ni *et al*., 2003). Quantitatively risk and severity can be measured by means of four attributes like exposure, likelihood and impact and mitigation process. The probability of security failure for each element in each scenario is estimated by means of building an attack graph and the severity of security failures for the same element (Pattanaik and Suhramaniam, 2011). The security failure analysis of the software creates another level of complications in identifying the important factors of the software process and their values to have a quantitative analysis. Hence, this work focuses on various regulatory noncompliance levels and criticality factors so as to determine the causality and controllability of various components in the six cause process model and the computations are carried out in fuzzy rough set space. The focus of the work is to study the impinging factors in the software applications that cause a security risk and threat using fuzzy rough set theory.

Software security risk, threat and vulnerability in a communication infrastructure: The software security is designed and realized by focussing on, the security risks, threats and their corresponding vulnerabilities with a communication infrastructure. The system software or the application software needs to be designed and developed after having considered all forms and levels of external attacks and insider leaks over the secured software development life cycle processes. The non-licensing, delayed renewal, non-updated and obsolete components or software libraries that are vulnerable to many attacks that lead to many security risks. The security features cannot be incorporated by introducing a mere set of modules at the coding phase. The uncertain pattern of attacks and possible vulnerabilities during state transfer and communication between physical or virtual modules are to be explored and analysed by a team of security experts and analysts for all possible risks and its security impacts (Sharif and Basri, 2001). Hence, the software security risk analysis can be grouped into five groups of activities:

- Defining risks, threats and vulnerabilities
- Automated risk detection through RACF
- Testing, identification of flaws and validation
- Reporting and generation of RASF
- Remediation and mitigation

Basically, risk is an uncertain event or a threatening agent or it is a combination of an abnormal event or failure and the consequence of the event leads to system damage or loss of life or injury (Mkpong *et al*., 2007). The risk analysis at the design level is an important part of software security and risk analysis process is a continuum and applied to different levels of software development life cycle to identify threats and vulnerabilities. All threats exploit vulnerabilities and causes security breach and vulnerability refers to weakness or openness to attack or it is a flaw in the system design.

The risks identified in the design and implementation stage in the software development life cycle are to be documented (Lee, 2011). The process of evaluating and identifying the risk is done by risk assessment model and then the identified risks are prioritized, mitigated and managed by risk acceptance model (Gilliam, 2004). Software is to be secure so that it is free from vulnerabilities and defects so as to continue and function correctly in spite of attack or misuse (Zhang *et al*., 2010).

MATERIALS AND METHODS

Proposed six cause process model: The controllability of a system states that the ability of a system to move from its initial state to final state with fixed interval of time. It refers to the stabilization of the system whereas the confidentiality prevents the sensitive information access must be restricted to the unauthorized users. Complexity is a condition or a situation with a higher degree of uncertainty.

Compliance is a set of standards and rules with safety and security regulations and it conforms to its stated security requirements whereas composability deals with the interrelation of components (Rehman and Mustafa, 2009). A highly composable system provides recombinant components that can be selected and assembled in various combinations to satisfy specific user requirements. Criticality refers to the critical state of the system.

Causality states that the relationship between a set of factors which are said to be causes and an outcome phenomenon called effects anything that affects an effect in the software is a factor of that effect. Controllability is a joint probability of all output parameters with respect to the input parameters under given conditions (Pearl, 2009).

The parameters in the risk controllability and security compliance are the causes of vulnerabilities in the software modules, interfaces, databases that lead to effects like security threats for the assets in the networking environment. Threats lead to the possibility of danger that might exploit vulnerability causes security breach (Tevis and Hanilton, 2004).

The inner safety triangle is visually represented by the vertices like compliance, controllability and criticality factors. The outer security triangle is visually represented by the vertices like complexity, composability and confidentiality factors. Lower safety approximation is bounded by the inner circle and it can be represented as Threat Free But Hazard (TFBH). Upper security approximation is bounded by the outer circle and it can be represented as Threat Hazard Both Free (THBF). All these factors are combined together to form the inner safety triangle and outer security triangle with lower safety approximation and upper security approximation regions which is described in Fig. 1.

The level of criticality in the safety triangle can be mathematically represented as the summation of all the probabilities of control and compliance of individual software components:

$$\begin{aligned}&\text{This can be written as} =\\&\Sigma\ \text{prob}\big(\text{criticality}(c_1,\ c_2,\ c_3...)\big) =\\&\text{prob}\big(\text{control }(\text{a1, a2, a3}\ldots)\big)\\&\text{AND prob}\big(\text{compliance}(\text{b1, b2, b3...})\big)\end{aligned} \tag{1}$$

$$\begin{aligned}&\text{Generically from the vertices of}\\&\text{inner safety triangle as, }P\big(\text{criticality}(c_k)\big) =\\&\Sigma\,\text{prob}\big(\text{control }a_j\big)\text{x prob(compliance}\big(b_j\big)\end{aligned} \tag{2}$$

Next, the confidentiality is a type of interval estimate for a parameter and it is used to indicate the reliability of an estimate. More specifically, the meaning of the term confidentiality refers to the confidence intervals are constructed across many separate data analyses of repeated and possibly different experiments, the proportion of such intervals that contain the true value of the parameter will match the confidence.

The design complexity of the individual software modules and the code complexity lead to the effective complexity of the software that is to be composed. Hence the composability is a system design principle that deals with the interrelationships of compound components with their containers. A highly composable system provides recombinant components that can be selected and assembled in various combinations to satisfy specific user requirements. Complexity is the inverse of stability and reachability.

The level of composability in the security triangle can be mathematically represented as the summation of all the probabilities of confidentiality and complexity of individual software components:

$$\begin{aligned}&\text{This can be written as} = \Sigma\,\text{prob}\big((\text{c1, c2, c3}\ldots)\big)\\&\text{prob}\big(\text{confidentiality}(\text{a1, a2, a3}\ldots)\big)\\&\text{AND prob}\big(\text{complexity}(\text{b1, b2, b3...})\big)\end{aligned} \tag{3}$$

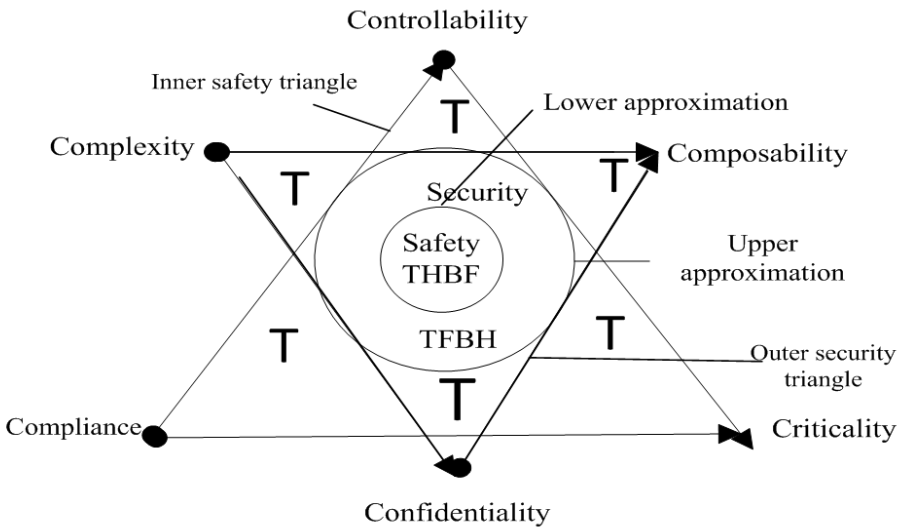


Fig. 1: Star of six cause in fuzzy rough set

Where:
Confidentiality = 1-Identifiability
Complexity = 1/stability×Reachability

In any systems with six causes, identifiability is a property with which a model must satisfy the process of drawing conclusion from data that are subject to random variables. Reachability refers to the ability to get from one node to another node within a graph or in a tree. Stability is the property that a random variable can take a set of possible different values due to the probability of occurrence or by chance. The system with software modules, the complexity is categorized into algorithm design complexity and code complexity whereas the primary problem of design complexity is that it has to be solved prior to the code implementation. Then the complexity of the software is analyzed at the design stage which permits the code is derived from implementation stage. Thus, it can be written as:

$$\begin{aligned}&\text{Software complexity} = n_c\\&\text{prob}(\text{reachability}) + (n_s + n_a)\text{Prob}(\text{stability})\end{aligned} \tag{4}$$

n_c represents the algorithmic or software complexity expressed as the difference between the number of nodes and edges, n_s represents the platform complexity expressed as the number of modules coupled within the framework and the new represents the infrastructure complexity expressed as the number of add-ons, library functions and hidden driver files. Generically from the vertices of outer security triangle in the star of six causes as:

$$\begin{aligned}&\text{Prob}\left(\text{compose}\left(c_{ij}\right)\right) = \sum \text{prob}\left(\text{confidentiality } a_j\right)\\&\text{x}\,\text{prob}\left(\text{complexity } b_j\right)\end{aligned} \tag{5}$$

$$\begin{aligned}&\text{Prob}\left(\text{compose}\left(c_{ijk}\right)\right) = \sum \text{prob}\left(1-\text{identifiability}_i\right)/\\&\text{prob}\left(\text{stability}_j\right)\text{x}\,\text{prob}\left(\text{reachability}\right)\end{aligned} \tag{6}$$

The features of a software system must comply with respect to various standards and multiple regulations. The software security standard ISO 19011:2011 provides an introduction to compliance auditing against various ISO management system standards. ISO 21827:2011 defines the essential characteristics of system security engineering and this model covers the entire system development life cycle. ISO/PAS 28000 standard specifies the requirement for security management systems. ISO 31000:2009 international standard recognizes the variety of the nature, level and complexity of risk and provides generic guidelines on principles and implementation of risk management. ISO/IEC 31010:2009 standard treats risk assessment as an integral part of risk management.

Proposed model of risk assessment: The cross behavior, functionality of the software, architectural design, standards of the software, risk policies of the software are taken as the input which is depicted in Fig. 2. The six cause such as controllability, complexity, criticality, confidentiality, compliance and composability are identified with respect to asset, attack and association then the risk of causal model is accepted through risk acceptance function and it is assessed by means of risk

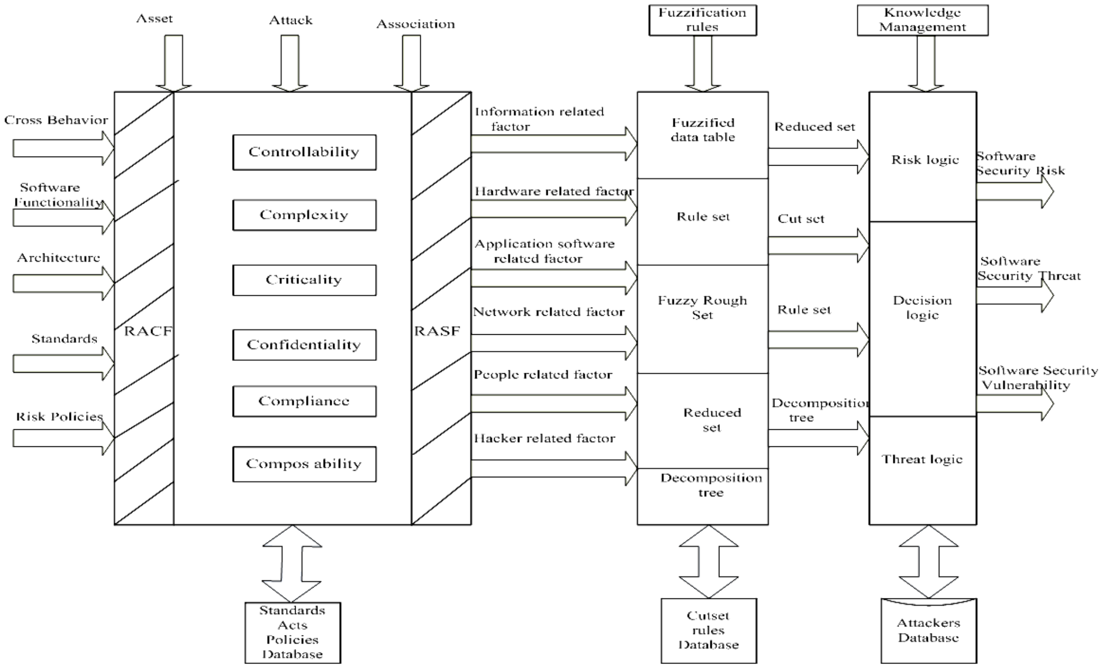


Fig. 2: Input-output-process model of risk assessment

assessment function. The outputs of the causes are subdivided into six factors like impact of information content loss, impact based on hacker activity, impact on hardware resource loss, impact on the application of software service loss, the impact of network authenticity loss and impact on social networking. The above six factors from the large massive database are fed as the input to the fuzzy rough set and it is fuzzified. The falsified data are categorized into reduced set, cut set, rule set and decomposition tree and it is processed through decision logic with software security risk software security threat and software security vulnerability to get as the output. The security risk of the software which is either in the form of an application hosted on a web or system software module configured towards a specific hardware and network that due to flaws or vulnerabilities in multiple levels. The factors that is responsible for the security feature of any software is mainly due to people who involved in the transactions, including direct and indirect users and the persons who are not actually involving in the legitimate transactions. There are some persons who are ready to intervene the regular operations and execution of software services from nowhere to gain some tangible benefits. Apart from these human interventions, the automated services may also be charged some responsibility in the leak of confidential information. The hardware parts from different companies with known configurations along with the chip sets are also responsible for the theft f information or data executed by the code segments. The network that is connecting these old and obsolete machines with limited resources to hold or block the critical data and the application itself may be designed in unsecured procedures hosted on an insecure server. All these factors can be represented as a pyramid indicating the reasons for the corresponding layer vulnerabilities as shown in Fig. 3. Figure 3 describes that the risk pyramid consists of six layers from bottom to top layers, namely hacker layer, user layer, hardware layer, network layer, application layer and the information layer. On the other side with compliance, controllability, criticality, complexity, composability and confidentiality.

The proposed six cause process model is compared with earlier models like octave, stride and square are compared with the proposed six cause risk analysis model and the comparison is elaborated in Table 1. The octave process model focuses on organization oriented activities and its governance results whereas the stride process model is activity oriented and its focuses on necessary goals of actions. Square process model focuses on goal oriented concepts and techniques of execution. The proposed six cause model focuses on cause oriented and the association between the causes and features. The octave process model performs risk based strategic assessment and planning technique for security. The stride process identifies the threat and performs threat

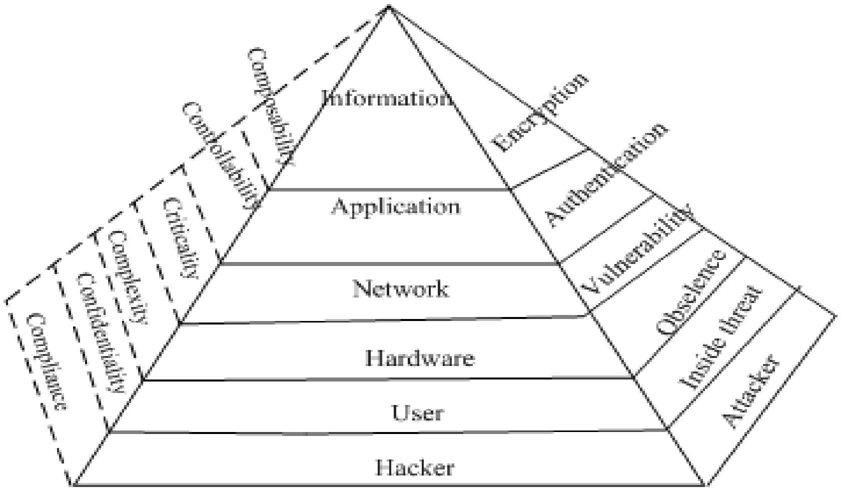


Fig. 3: Risk pyramid

Table 1: Comparison of software security process model and proposed process model confidentiality

OCTAVE process	STRIDE process	SQUARE process	Six causes process
Focus	Focus	Focus	Focus
Organization oriented and its activities, governance results	Activity oriented and the necessary goals of actions	Goal oriented through concepts and techniques of execution	Cause oriented and the association between the causes and features
Objective and plans:	Objectives and plans:	Objectives and plans:	Objectives and plans:
Performs risk based strategic assessment and planning techniques for security	Identifies the threat And performs threat modeling	Build security concepts in the early stage of SDLC and Identifies security goals	Identification of process and product level regulatory non compliances that result risks
Methods:	Methods:	Methods:	Methods:
Operationally Critical Threat Asset and Vulnerability Evaluation	Spoofing identity Tampering with data Repudiation Information disclosure Denial of service Elevation of privilege	Security engineering Quality engineering Requirements engineering methods	Controllability measuring Complexity factors Criticality identification Confidentiality declaration Compliance checking Composability alerting
Design strategy:	Design strategy:	Design strategy:	Design strategy:
It focuses on the design and regulations for the strategic planning of the organization	It performs system evaluation of threat	Design and develop essential artifacts	Analyze the vulnerabilities due to legal and applications deployed in specific environments
Limitations:	Limitations:	Limitations:	Added features:
OCTAVE is not suitable for Ad-Hoc vulnerability assessment and performs techniques for risk analysis towards security organizational evaluation not on legal compliances	STRIDE helps to identify threats to all security goals, including confidentiality, integrity, availability, authentication, authorization and non-repudiation not on regulatory compliance	SQUARE perform risk assessments only Selective elicitation techniques and Categorize requirements not on software functional, legal and regulatory requirements	The six causes process model is a risk assessment model as a part of security requirement management in terms of the probable causes and effects
Weightage:	Weightage:	Weightage:	Weightage:
It is an asset driven evaluation approach For risk analysis of the organizational assets	Threats are identified by applying STRIDE to all system functions	Prioritize requirements With a requirements inspection	The six causes process helps to perform risk identification, risk acceptance and risk assessment of security requirements
Attributes and outputs:	Attributes and outputs:	Attributes and methods:	Attributes and methods:
Analysis team Generic threat profile Evaluation activities Focus on risk	Identify security objective Identify vulnerability Identify threat	Identify security goals Develop artifacts Perform risk assessment Categorize and prioritize security requirement	Risk acceptance function Risk assessment function Reduced relations

modeling whereas the square process model builds security concepts in the early stage of the software development life cycle. The six cause process model identifies the process and product level regulatory non-compliances. The proposed model incorporates factors like controllability, complexity factors, criticality identification factor, confidentiality declaration factors, compliance checking and composability alerting factors.

The octave process model focuses on design and strategic planning of the organization whereas stride focuses on evaluation of threat and square focuses on the design and development of essential artifacts. The proposed six cause model also analyzes the vulnerabilities due to legal and regulations for the applications deployed in a specific environment. The limitation of the earlier octave process model that it is not suitable for Ad-hoc vulnerability assessments.

The stride process model cannot identify the security threats in non-repudiation attack and it is not suitable for regulatory compliance whereas the square process model performs risk assessment only on selective elicitation techniques. The octave model performs generic threat profile and evaluation activities as the output whereas the stride process model identify threat, vulnerability and security objective as output. The proposed six cause process model has a risk acceptance function to accept the risk and it has a risk assessment function to access the accepted risk and then the accepted risk had been reduced and managed it focusing on the causes and effects of risk.

The six cause software security process model can be considered as a quantitative model emerged from their respective attributes of the various causes. The relationship between various causes are studied and correlated with each other to form a fuzzy rough set. The rough set as it implies the lower and upper bound on the values of the attributes for the factors are summarized. The priorities and weightings are assigned based on the application execution environment in order to identify the cut set of reasons and their relationships with each other through a decomposability tree. The acceptance function and assessment function are declared within the fuzzy computing framework and the corresponding parameters are passed into the function as arguments.

The exiting communication infrastructure security and its risk analysis are carried out based on the three Octave, Stride and Square approaches. There is a need for understanding the various causes for all the activities covering the governance and the goal of the security mechanisms deployed in the infrastructure or within the data communication network. The different types of vulnerabilities and attacks with their relative weightings are considered for the research work. The various assets in the data communication networks and the priority of information packets are considered to arrive at a most suitable mechanism to identify the cause and the impact of the attack which are uncertain. The fundamental challenge in any risk management problem is to identify the risk and also assert that the incident will lead to potential risk in the software functionality or any other quality parameters. Then only it is possible to monitor measure and mitigate the risk within the available networked resources. The risk mitigation needs decomposition and minimum and optimal way of solving the issues in the security domain. The proposed six cause model finds solution for the risks to be identified, measured and minimized to take remedial action when treating all the possible reasons as fuzzy values. The fuzzy rough set approach is one of the key computing methodologies where in these situations are handled and explored.

The quantification of the effect and impact by any forms of attacks and the remedial actions is justified using minimum rules. The various scenarios with multiple causes and different forms of vulnerabilities are introduced and the correct cause with and acceptance function is determined by the proposed model. The impact being a superset of the end result in the security domain is portioned into deterministic threat and probabilistic risk in the proposed context. The serious limitation of the reserach is the lack of exact technical specifications of various networks considered and can be implemented as virtual privat networks for further research study.

This analysis and strategy can be extended to any form of network device and mobile networking with corresponding types of security flaws and threats possibility. The matrix driven model gives the extensibility and scalability of the model when more types of networks are considered as a communication infrastructure.

Proposed risk acceptance model: The security risk can be expressed as the potential of expectation of any attack among the many choices of the incidents and the influential factor for the attack to happen or vulnerability and the amount of desire of the specified attack on the fixed asset. This is expressed as:

$$\text{Risk Potential} = \text{Expect}\left[\{C\}\times\{I\}\times\{D\}\right] \tag{7}$$

The term Expect represents the expectation values of the outcome or income value of the content. C represents a set of choices from many and I represent the sign of influential value either positive or negative or both as unsigned D represents the type of desire as willingness or unwillingness or both types.

The operations performed by a hacker for example, can be considered as set A that represents the nature of the activity and the set S represents the status of the actions or inactions performed by the attacker. The hacker might have attacked for a short duration with severe malware or a virus on the specific server but not steal the data fully. The set E represents the timing, location and environment of the activity:

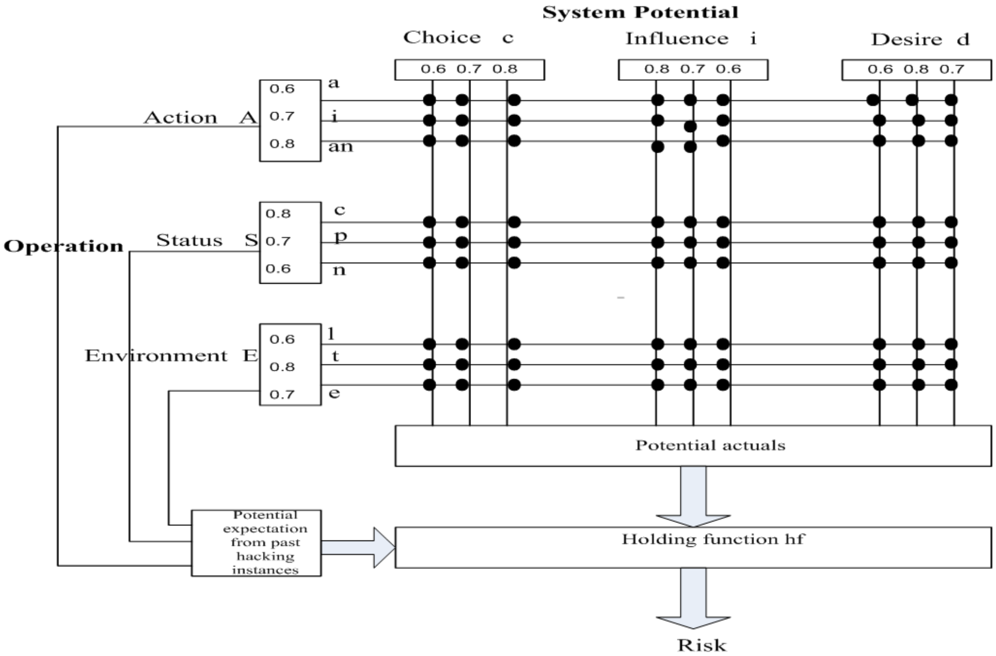


Fig. 4: Risk acceptance analysis

- {A} = {act, inact, antiact} = {a, o, a'}
- {S} = {complete, partial, null} = {cp, pa, no}
- {E} = {location, timing, environment} = {loc, Time, env}

This proposed risk acceptance model is depicted in Fig. 4 where the intersection points between the expectation from the potential of the underlying system and the operational factors in the context of security are shown as acceptable risky instances.

The total possibilities of such meeting points in the model are nine sets of nine coincidences which form nine sets of matrices. The risk potential actual is obtained, then it is compared with past hacker operation the difference between the potential expected and the actual is processed by holding function h_r then the resultant risk is accepted by risk acceptance function:

$$\text{Risk Acceptance} = \left| \begin{matrix} \text{Risk potential expected} - \\ \text{Risk potential actual} \end{matrix} \right| \quad (8)$$

RESULTS AND DISCUSSION

Rough set analysis of risk acceptance matrix: The risk acceptance matrix is used to define the various levels of risks. The risk matrix is used to identify the severity of the event and the probability of occurrence of the event:

$$\begin{bmatrix} A_1C_1 & A_1C_2 & A_1C_3 \\ A_2C_1 & A_2C_2 & A_2C_3 \\ A_3C_1 & A_3C_2 & A_3C_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0.36 & 0.42 & 0.48 \\ 0.42 & 0.49 & 0.56 \\ 0.48 & 0.56 & 0.64 \end{bmatrix}$$

a_1 c_1 holds the value 0.36 (a1 = 0.6 and c_1 =0. 6) Whereas a_2 c_2 holds the value 0.49 (a_2 = 0.7 and c_2 = 0. 7):

$$\begin{bmatrix} S_1d_1 & S_1d_2 & S_1d_3 \\ S_2d_1 & S_2d_2 & S_2d_3 \\ S_3d_1 & S_3d_2 & S_3d_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0.48 & 0.64 & 0.56 \\ 0.42 & 0.56 & 0.49 \\ 0.36 & 0.48 & 0.42 \end{bmatrix}$$

s_1 d_1 holds the value 0.48 (s_1 = 0.8 and d_1 = 0. 6) Whereas s_2 d_2 holds the value 0.56 (s_2 = 0.7 and d_2 = 0.8):

$$\begin{bmatrix} E_1i_1 & E_1i_2 & E_1i_3 \\ E_2i_1 & E_2i_2 & E_2i_3 \\ E_3i_1 & E_3i_2 & E_3i_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0.36 & 0.48 & 0.42 \\ 0.48 & 0.64 & 0.56 \\ 0.42 & 0.56 & 0.49 \end{bmatrix}$$

E_2 i_2 holds the value 0.64 (E_2 = 0. 8 and i_2 = 0.7) whereas E_3 i_3 holds the value 0.49 (E_3 = 0. 7 and i_3 = 0.7). Similarly, the risk acceptance matrix is constructed for action, status and environment with respect to choice, influence and desire. Risk potential is an unsigned value derived from the many choices of attacks in the risk context during the time with the content responsible for the damage and cost. For example, the risk potential of a spoofing attack among the multiple choices of attacks that damage the integrity in the information security environment as the second time of its occurrence by changing the content that is address value of the customer towards a maximum business gain. This can be generalized as an expression given below: For example: RACF = {[CHOICE. k(CONTEXT. Time (CONTENT. Low (money))]} >= 0.4 = Otherwise NOT accepted as risk If the choice is k out of n and the context is c out of d at the incident time t, when the content is a string of characters of length l the effect of this cause is 10000$ then the holding function $h_f{}^r$ = 0.3 In the second case the security risk is termed as:

$$\text{Security risk} = h_f^r \times \left(\text{Pot}_{\text{exp}} - \text{Pot}_{\text{actual}}\right) \tag{9}$$

$$\begin{gathered}\text{Risk acceptance function} = \text{choice}\left(\text{act} - \text{inact}\right) + \\ \text{influence}\left(\text{partial}\right) + \text{desire}\left(\text{time} + \text{location}\right)\end{gathered} \tag{10}$$

As an illustration, consider choice =0. 6, influence = 0. 8, desire = 0. 7, A_1 c_1 = 0. 36, A_3c_1 = 0. 48, S_2 i_1 = 0. 56, E_1 d_3 = 0. 42 and E_2 d_3 = 0. 56 then the actual potential would be actual potential = 0.6 (0.36-0.48)+0.8 (0.56)+0.7 (0.42+0.56). Actual potential = 1.014. Consider the potential expected maybe 1.05 then the security risk is calculated as. Security risk = 0.3×(1.05-1.014). Security risk = 0.0108 to 81 security breach instance or operations. So there should be a security risk is about 0.013% for 81 operations for 1000 operations it may be 0.00108%.

As a second scenario if the choice = 0. 7, influence = 0. 6, desire = 0. 8, A_1 i_2 = 0. 42, A_3 i_2 = 0. 56, S_2 i_3 = 0. 42, E_1 d_2 = 0. 48 and E_2 d_2 = 0. 64 then the actual potential would be Actual potential = 0.7 (0.42-0.56)+0.6 (0.42)+0.8 (0.48+0.64) Actual potential = 1.05 Consider the potential expected maybe 1.085, the security risk is calculated as. Security risk =0. 3* (1.085-1.05) Security risk = 0.0105-81 security breach instance or operations. So, there should be a security risk of 0.0129% for 81 operations and it will become 0.001% for 1000 operations.

Software security risk classifications based on attacks towards risk assessment: Software security risk assessment can be done by classifying the risks based on the nature of attacks. The total number of packets received out of the number of packets transmitted can be represented in terms of percentage. So, the impact of the security risk assessment can be done in terms of percentage. Spoofing provides false information about a principal identity to obtain unauthorized access to systems and their services. It modifies the source address of the packet making network to fail. So the impact of information content loss in percentage may be taken as 98% whereas the impact of application software service loss may be taken as 72 %, the impact of network authentication loss is 98%, impact on people or social networks is 42%, impact of hardware resource loss is 72% and impacts based on hacker activity is 92%. In Tampering of data integrity authorized parties are able to modify data Access rights and privileges can be modified (Russell, 2002). So, the impact of information content loss in percentage is 84% whereas the application software service loss in percentage is 74%, the impact of network authentication loss in percentage is 65%, impact on people or social networks is 68%, impact of hardware resource loss is 84% and impacts based on hacker activity is 79%. Buffer overflow crashes the target program. Buffer overflow occurs when software fails to check the input it's been given. So, its impact on information loss in percentage may be taken as 96% whereas the impact on the application of software service loss in percentage may be taken as 92%, the impact of network loss in percentage is 74%, impact on people or social networks is 69%, impact of hardware resource loss in percentage is 92% and impacts based on hacker activity is 91%.

The sniffer is an application that can capture network packets. Sniffers are also known as network protocol analyzers. While protocol analyzers are really networking, troubleshooting tools, they are also used by hackers for hacking network. If the network packets are not encrypted, the data within the network packet can be read using a sniffer.

Sniffing refers to the process used by attackers to capture network traffic using a sniffer. Once, the packet is captured using a sniffer, the contents of packets can be analyzed. Sniffers are used by hackers to capture sensitive network information, such as passwords, account information, etc. (www.omnisecu.com/security). So, the impact of information content loss in percentage may be taken as 98% whereas impact on the application of software service loss may be taken as 0, impact on network authentication loss in percentage is 82%, impact on people or social network re is 0, impact of hardware resource loss is 82% and impacts based on hacker activity in percentage is 69%.

Hacker attack exploits weakness in the computer system whereas it reveals the administrator credentials and it manipulates the behavior of network connection (www.techopedia.com/definition) so the impact of information content loss in percentage is 97%, impact on application of software service loss is 0, impact of network authentication loss in percentage is 95%, impact on people or social networking is 94%, impact on hardware resource loss is 63% and impact of hacker activity is 98%. ARP positioning attack sends fake message on to local

Table 2: Fuzzified data table

Security attack Activity	Impact on information content loss in percent	Impact on application of software service loss in percent	Impact on the network loss authenticity in percentage	Impact on people or social networking loss in percent	Impact on hardware resource loss in percent	Impact based on hacker activity loss in percent
Spoofing	98	72	98	42	72	92
Tampering-data integrity	84	74	65	68	84	79
Non reputation	72	84	98	90	0	74
Denial of service	98	72	98	42	72	92
Eavesdropping	84	74	65	68	84	79
Phishing	83	0	69	73	0	83
Malware	91	72	0	65	74	93
Buffer overflow	96	92	74	69	92	91
Hijacking	97	0	95	94	62	98
Password attack	72	0	92	65	42	78
Exploit attack	96	92	74	69	92	91
Information disclosure	72	84	98	90	0	74
Man in the middle attack	96	92	74	79	92	91
Sniffer attack	98	0	82	0	82	69
Hacker attacks	97	0	95	94	62	98
DNS server spoofing	96	92	74	69	92	91
ARP positioning	83	0	69	73	0	83
Ping of death	98	0	82	0	82	69
Session hijacking	91	72	0	65	74	93
Smurf attack	72	0	92	65	0	78

area network so the impact of information content loss in percentage is 83%, impact on application of sofware service loss is 0, impact on network authentication loss is 69%, impact on people or social network is 73%, impact related to hardware resource loss is 0 and its impact based on hacker activity is 83%. Session hijack attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server.

So, the impact of information content loss in percentage may be taken as 91% whereas the impact on application of software service loss may be taken as 72%, impact on network authentication loss in percentage is 0, impact on people or social network is 65%, impact on hardware resource loss in percentage is 74% and impact based on hacker activity is 93%. Smurf attack is a type of denial of service attack in which a system is flooded with spoofed ping messages. This creates high computer network traffic on the victim's network which often renders it unresponsiveness (www.oswap.org).

So, the impact of information content loss in percentage may be taken as 72% whereas the impact on the application of software service loss may be taken as 0, impact on network authentication loss is 92%, impact on people or social networking is 65% and impacts on hardware resource loss is 0 and impacts based on hacker activity is 78% (Pistoia *et al*., 2004) which is described in Table 2.

Risk assessment using a fuzzy rough set model of software security risk: Risk may be assessed in terms of the quantifiable likelihood of loss or less-than-expected returns in business and or technical terms, overcoming the threats from outside and inside over the existing vulnerabilities within the unprotected physical and computational environment.

A discernibility matrix of a decision table D = (U, C∪D) is a symmetric |U|×|U| matrix with entries defined as:

$$d_{ij} = \{a \in C \,|\, a(x_i) \neq a\ (x_i)\}\ i,\ j = 1, |U\ |\} \qquad (11)$$

Each d_{ij} contains those attributes that differ between objects i and j. Fuzzy Rough Set Feature Selection (FRFS) addresses these problems and retains dataset semantics. FRFS is applied to two challenging domains where a feature reducing steps is important, namely, web content classification and complex system monitoring. The fuzzy P-Upper and P-Lower approximation are defined as:

$$\mu \underline{PX}(x) = \sup_{F \in U} \min\left(\mu F(x), \inf_{y \in U} \max\left\{ \begin{matrix} 1- \\ \mu F(y) \end{matrix}, \mu X(y) \right\}\right) \qquad (12)$$

$$\mu \overline{P}X(x) - \sup_{F \in U/P} \min\left(\begin{matrix} \mu F(x), \\ \sup_{y \in U} \min\{\mu F(y) \end{matrix}, \mu X(y)\}\right) \qquad (13)$$

Fuzzy rough sets are the generalization of the traditional rough set to deal with both fuzziness and vagueness in data. Security risk is assessed through the fuzzy rough set and it consists of risk factor table, reduced set, rule set, cut set, decomposition tree and result described in Fig. 5.

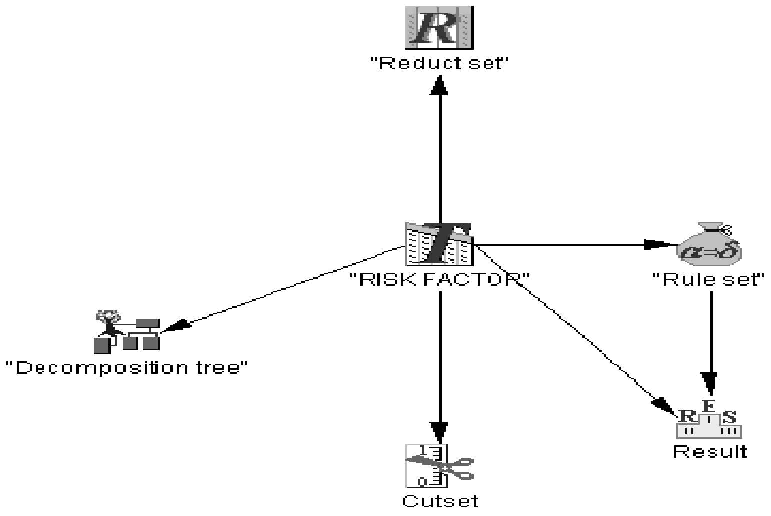


Fig. 5: Security attack risk as rough sets

The risk factor table consists of six attributes and twenty objects obtained from Table 2 whereas it consists of the fuzzified data values for the various security attacks with respect to the impact of information content loss in percentage, impact on the application of software service loss in percentage, the impact of network authenticity loss in percentage, impact on people or social networking loss in percentage, impact of hardware resource loss in percentageand impact based on hacker activity loss in percentage.

Table 3 describes the rule set for fuzzy risk in security attack whereas 41 rules are obtained and match represents the number of matches with the decision rule. The decision rules and the matches are obtained based on the values of the attributes mentioned in fuzzified data table. The reduced set of fuzzy risk in a security attack describes that of identification number from 1-8, the first row of size 2 with two attributes, namely the impact of information content loss in percentage and impact on the application of software service loss whereas the impact of information content loss in percentage is a subset of impact on the application of software service loss, likewise all the attributes relationship is described in Table 4.

Risk factor decomposition tree analysis: The risk factors due to various vulnerabilities and attacks are to be classified and identified by means of process decomposition whereas the decomposition tree decomposes large data set into fragments then the fragments are further decomposed into nodes and leaf nodes. According to the maximum size of the leaf the algorithm generates the subsequent level based on the condition, each node in the decomposition tree have a template and associated subset of training samples, based on the samples the decision rules are generated for the corresponding leaf node.

The root node is decomposed into two nodes the left node is having the impact of information content loss ≥96 whereas it is further decomposed into two leaf nodes. The left leaf node is having the ("impact of information content loss"≥96) and ("impact of information content loss"≥98) whereas the size of leaf with four matches and two decision rules are:

- ("Impact on the application of software service loss" = 72) and ("impact of information content loss" = 98) (("impact based on hacker activity" = {92}
- ("Impact on the application of software service loss" = 0) and ("impact of information content loss" = 98)_("impact based on hacker activity" = {69}

The right leaf node is having the ("impact of information content loss "≥96) and ("impact of information content loss "<98) than the size of the leaf with six matches and two decision rules are:

- ("Impact of information content loss "= 96) and ("impact on the application of software service loss = 92)_("impact based on hacker activity" {91}
- ("Impact of information content loss" = 97) and ("impact on the application of software service loss = 0") and _("impact based on hacker activity" = {98}

Table 3: Rule sets for fuzzy risk in security attack

Match	Decision rules
4	(“Impact of information content loss” = 96)¬(“impact based on hacker activity” = {91 (Daniel *et al.*, 2005)})
4	(“Impact on the application of software service loss” = 92)¬(“impact based on hacker activity” = {91 (Daniel *et al.*, 2005)})
4	(“Impact on network authenticity loss” = 74)¬(“impact based on hacker activity” = {91 (Daniel *et al.*, 2005)})
4	(“Impact on hardware resource loss” = 92)¬(“impact based on hacker activity” = {91 (Daniel *et al.*, 2005)})
3	(“Impact on people or social networking” = 69)¬(“impact based on hacker activity” = {91 (Pattanaik and Subramaniam, 2011)})
2	(“Impact of information content loss” = 98) and (“impact on the application of software service loss” = 72)¬(“impact based on hacker activity” = {92 (Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 98) and (“impact on network authenticity loss” = 98)¬(“impact based on hacker activity” = {92 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 42)¬(“impact based on hacker activity” = {92 Allen *et al.*, 2008)})
2	(“Impact on hardware resource loss” = 72)¬(“impact based on hacker activity” = {92 Allen *et al.*, 2008)})
2	(“Impact on the application of software service loss” = 72) and (“ impact on network authenticity loss” = 98)¬(“impact based on hacker activity” = {92 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 84)¬(“impact based on hacker activity” = {79 Allen *et al.*, 2008)})
2	(“Impact on the application of software service loss” = 74)¬(“impact based on hacker activity” = {79 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 65)¬(“impact based on hacker activity” = {79 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 68)¬(“impact based on hacker activity” {79 Allen *et al.*, 2008)})
2	(“Impact on hardware resource loss” = 84)¬(“impact based on hacker activity” = {79 Allen *et al.*, 2008)})
2	(“Impact on the application of software service loss” = 84)¬(“impact based on hacker activity” = {74 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 72) and (“impact on network authenticity loss” = 98)¬(“Impact based on hacker activity” = {74 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 90)¬(“impact based on hacker activity” = {74 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 98) and (“impact on hardware resource loss” = 0)¬(“impact based on hacker activity” {74 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 83)¬(“impact based on hacker activity” = {83 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 69)¬(“impact based on hacker activity” = {83 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 73)¬(“impact based on hacker activity” = {83 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 91)¬(“impact based on hacker activity” {93 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 0)¬(“impact based on hacker activity” = {93 Allen *et al.*, 2008)})
2	(“Impact on the application of software service loss” = 72) and (“Impact on people or social network” = 65)¬(“impact based on hacker activity” = {93 [2]})
2	(“Impact on hardware resource loss” = 74)¬(“impact based on hacker activity” = {93 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 97)¬(“impact based on hacker activity” {98 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 95)¬(“impact based on hacker activity” = {98 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 94)¬(“impact based on hacker activity” = {98 Allen *et al.*, 2008)})
2	(“Impact on hardware resource loss” = 62)¬(“impact based on hacker activity” = {98 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 72) and (“impact on the application of software service loss” = 0)¬(“impact based on hacker activity” = {78 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 92)¬(“impact based on hacker activity” = {78 Allen *et al.*, 2008)})
2	(“Impact on the application of software service loss” = 0) and (“ impact on people or social networking” = 65)¬(“impact based on hacker activity” = {78 Allen *et al.*, 2008)})
2	(“Impact of information content loss” = 72) and (“impact on people or social networking” = 65) (“impact based on hacker activity” = {78 Allen *et al.*, 2008)})
2	(“Impact of information content loss”=98) and (“impact on the application of software service loss” = 0)¬(“impact based on hacker activity” = {69 Allen *et al.*, 2008)})
2	(“Impact on network authenticity loss” = 82)¬(“impact based on hacker activity” = {69 Allen *et al.*, 2008)})
2	(“Impact on people or social networking” = 0)¬(“impact based on hacker activity” = {69 Allen *et al.*, 2008)})
2	(“Impact on hardware resource loss” = 82)¬(“impact based on hacker activity” = {69 Allen *et al.*, 2008)})
1	(“Impact on hardware resource loss” = 42)¬(“impact based on hacker activity” = {78 (Sharif and Basri, 2001)})
1	(“Impact on people or social networking” = 79)¬(“impact based on hacker activity” = {91 (Sharif and Basri, 2001)})
1	(“Impact on people or social networking” = 65) and (“impact on hardware resource loss” = 0)¬(“impact based on hacker activity” = {78 (Sharif and Basri, 2001)})

Table 4: Reduced set of fuzzy risk in security attack

Size of the reduced attributes	Positive region	Scalability coefficient	Reduced set of rules
2	1	1	{”Impact of information content loss”, “impact on application of software service loss”}
2	1	1	{”Impact of information content loss”, “impact on network authenticity loss”}
2	1	1	{”Impact of information content loss “Impact on people or social network”}
2	1	1	{”Impact on application of software service loss”, “impact on network authenticity loss”}
2	1	1	{”Impact on network authenticity loss”, “impact on people or social networking”}
2	1	1	{”Impact on application of software service loss”, “impact on people or social networking”}
2	1	1	{”Impact on network authenticity loss”, “impact on hardware resource loss”}
2	1	1	{”Impact on people or social networking”, “impact on hardware resource loss”}

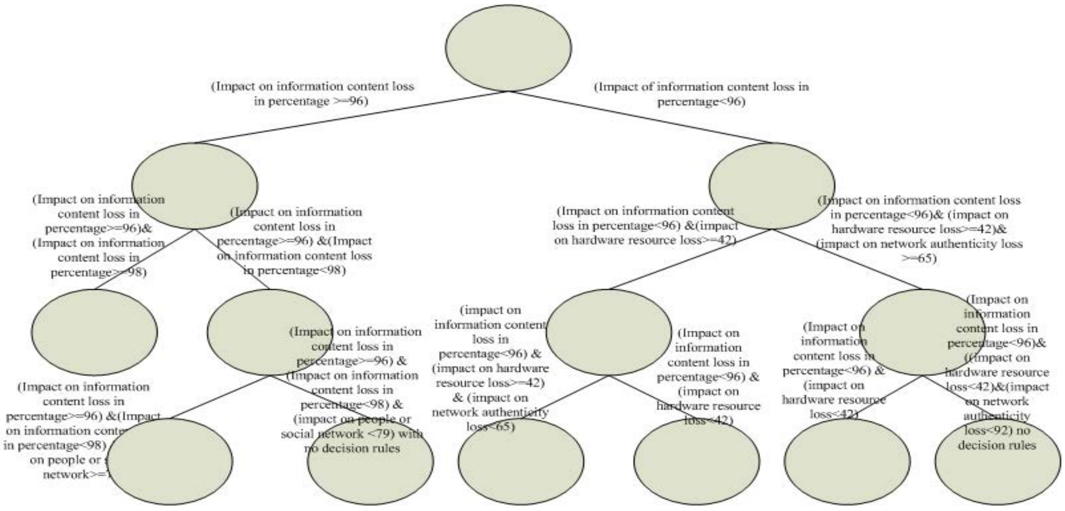

(Impact on information content loss in percentage >=96)
(Impact of information content loss in percentage<96)
(Impact on information content loss in percentage>=96)& (Impact on information content loss in percentage>=98)
(Impact on information content loss in percentage>=96) &(Impact on information content loss in percentage<98)
(Impact on information content loss in percentage<96) &(impact on hardware resource loss>=42)
(Impact on information content loss in percentage<96)& (impact on hardware resource loss>=42)& (impact on network authenticity loss >=65)
(Impact on information content loss in percentage>=96) &(Impact on information conte in percentage<98) on people or network>=
(Impact on information content loss in percentage>=96) & (Impact on information content loss in percentage<98) & (impact on people or social network <79) with no decision rules
(impact on information content loss in percentage<96) & (impact on hardware resource loss>=42) & (impact on network authenticity loss<65)
(Impact on information content loss in percentage<96) & (impact on hardware resource loss<42)
(Impact on information content loss in percentage<96) & (impact on hardware resource loss<42)
(Impact on information content loss in percentage<96)& ((impact on hardware resource loss<42)&(impact on network authenticity loss<92) no decision rules

Fig. 6: Decomposition tree with 13 nodes

Table 5: Analysis of decomposition tree

Total no. of nodes in the tree	Total no. of matches out of 41 rules	Level of the decomposition tree	Remarks of the risk analysis with respect to decomposition of attacks
3	20	1	Maximum no of rules obtained when tree Level decreases
7	18	2	Maximum no of rules obtained when tree Level decreases
9	14	2	Maximum no of rules obtained when tree Level decreases
13	10	3	Minimum no of rules obtained when tree level increases
15	6	3	Minimum no of rules obtained when tree level increases
21	No rule matches	4	There is no rule match when there is maximum tree level

The Right node with (“impact of information content loss “<96) having the left leaf node with (“impact of information content loss “<96) and (“Impact on hardware resource loss” ≥42) whereas the size of the leaf with four matches and two decision rules are:

- (“Impact of information content loss” = 84) and (“impact on hardware resource loss” = 84)_(“impact based on hacker activity” = {79}
- (“Impact of information content loss” = 91) and (“impact on hardware resource loss” = 74)_(“impact based on hacker activity” = {93}

The right leaf with (“impact of information content loss “<96) and (“Impact on hardware resource loss”<42) whereas the size of the leaf with four matches and two decision rules are:

- (“Impact of information content loss” = 72) and (“impact on hardware resource loss” = 0)_(“impact based on hacker activity” = {74}
- (“Impact of information content loss” = 83) and (“impact on hardware resource loss” = 0)_(“impact based on hacker activity” = {83} which is described in detail in Fig. 6

The analysis of decomposition tree in Table 5 describes the level of decomposition based on the number of nodes and the remarks of the risk analysis with respect to decomposition of attacks.

The risk assessment function using a fuzzy rough set model of software security risk is described by means of RASF whereas. Risk Assessment Function (RASF) = criticality [(complexity * confidentiality) * (1-compliance)]. Controllability * composability. Business Money transaction approval module has a very high criticality, the high causality, a high confidentiality with maximum compliance, but a low controllability and a very low composability.

The risk is calculated with respect to the six cause factors like controllability, confidentiality, complexity, compliance, composability and criticality is represented in Table 6. In Fig. 7-12 indicates the maximum risk factor of the system with respect to the complexity of secured component, criticality of the secured modules, compliance of software modules, the composability factor of security modules, controllability of secured modules, confidentiality of secured components. The risk assessment document is to identify the risk associated with a task, activity or process, then it prioritizes the risk, determine the hazard control measure, implement controls,

Table 6: Causes of risk

Controllability	Confidentiality	Complexity	Compliance	Composability	Criticality	Risk
0.8	0.9	0.3	0.4	0.5	0.6	0.28
0.5	0.1	0.2	0.8	0.8	0.7	0.1
0.3	0.5	0.7	0.6	0.8	0.1	0.1
0.9	0.7	0.6	0.5	0.3	0.4	0.24
0.5	0.3	0.2	0.7	0.8	0.6	0.1
0.6	0.4	0.5	0.8	0.2	0.5	0.41
0.7	0.8	0.9	0.5	0.3	0.7	0.24
0.2	0.4	0.5	0.6	0.7	0.9	0.64
0.3	0.1	0.2	0.5	0.6	0.7	0.1
0.4	0.6	0.7	0.9	0.2	0.1	0.15
0.2	0.1	0.5	0.4	0.5	0.7	0.17
0.3	0.1	0.4	0.5	0.6	0.4	0.1
0.1	0.7	0.8	0.4	0.7	0.5	0.8
0.3	0.2	0.1	0.5	0.2	0.1	0.1
0.5	0.7	0.6	0.8	0.9	0.7	0.26
0.3	0.2	0.4	0.1	0.6	0.4	0.1
0.6	0.8	0.7	0.9	0.4	0.5	0.35
0.3	0.1	0.3	0.4	0.5	0.7	0.1
0.4	0.3	0.1	0.2	0.6	0.5	0.1
0.2	0.8	0.7	0.6	0.5	0.4	0.6

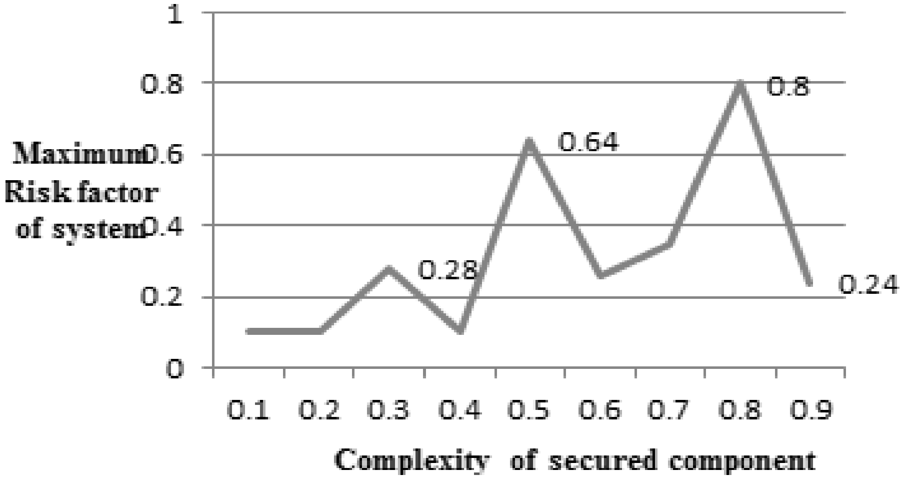


Fig. 7: Complexity of secured component vs. Maximum risk factor of system

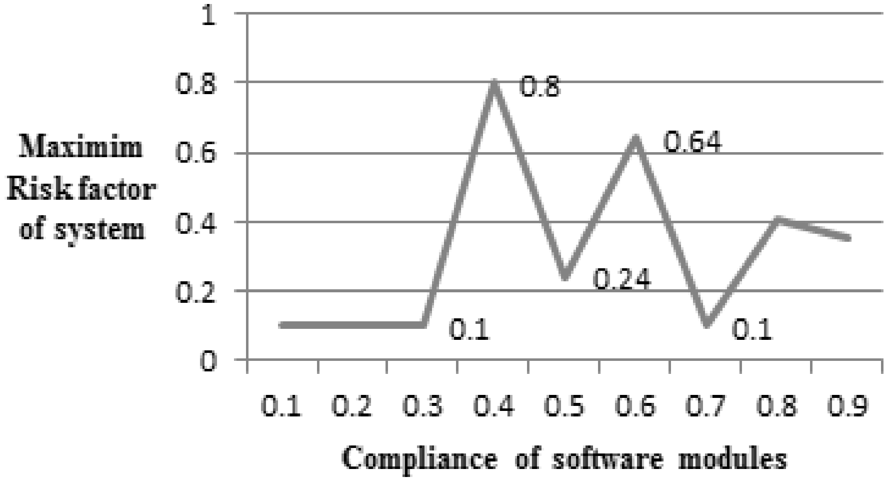


Fig. 8: Compliance of software modules vs maximum risk factor of system

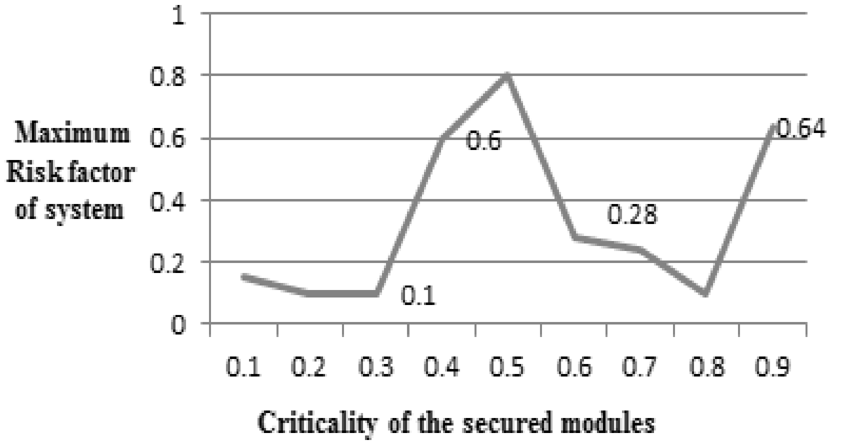


Fig. 9: Criticality of the secured modules vs maximum risk factor of system

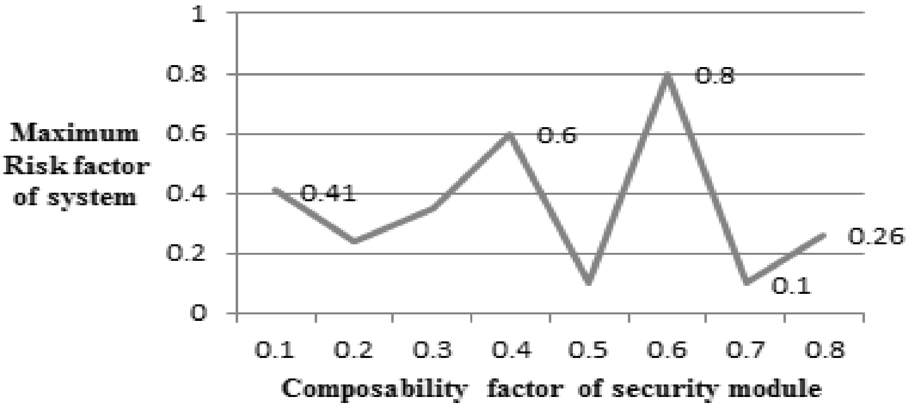


Fig. 10: Composability factor of security vs maximum risk factor of system

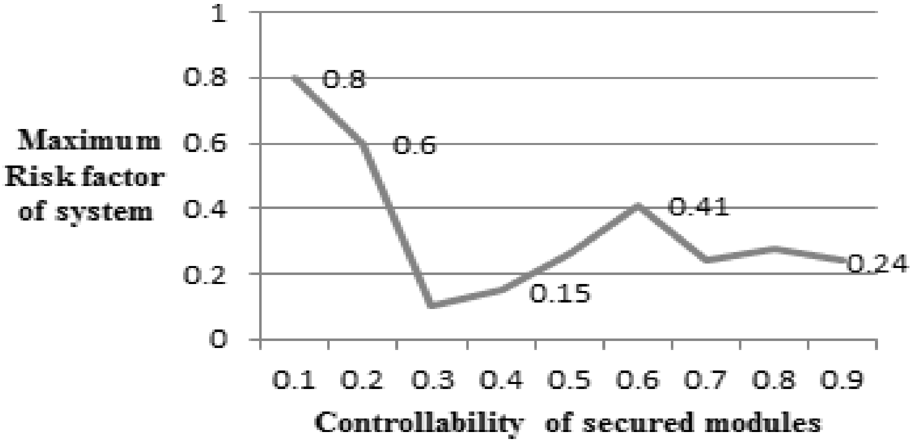


Fig. 11: Controllability of secured module vs maximum risk factor of system

measure the effectiveness of control and make changes to improve continuously. The risk assessment document is used to evaluate the risk and this procedure is carried out by a person experienced with the activity or process which is described in Fig. 13. The reduced set is obtained from the rule set attribute reduction is done in the reduced set and the result of attributes from the reduced set is described in Fig. 14.

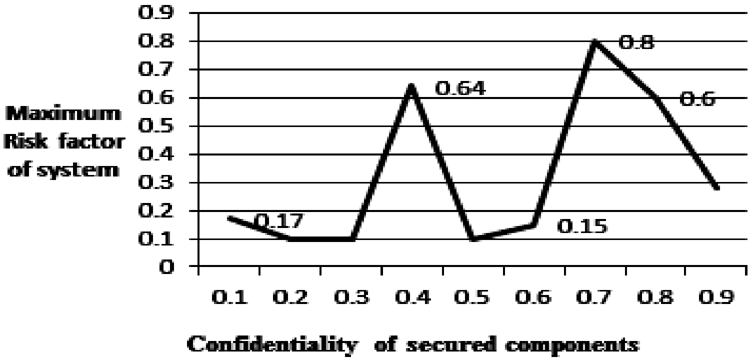
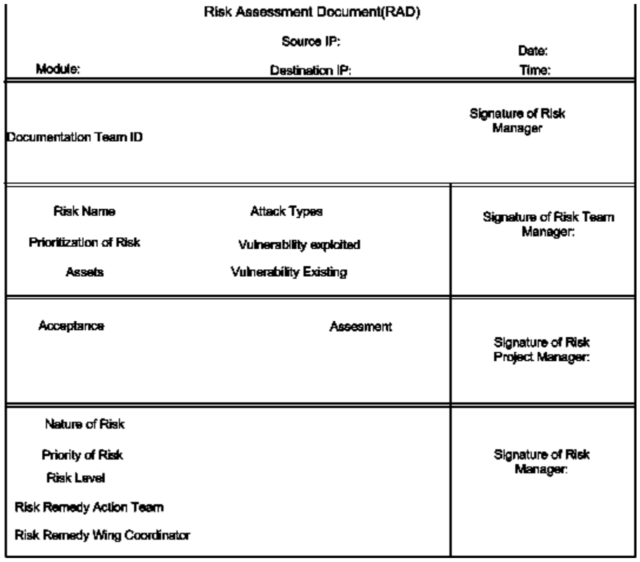


Fig. 12: Confidentiality of secured component vs maximum risk factor of system



Risk Assessment Document(RAD)

Source IP:

Date:

Module:

Destination IP:

Time:

Documentation Team ID

Signature of Risk Manager

Risk Name

Attack Types

Prioritization of Risk

Vulnerability exploited

Assets

Vulnerability Existing

Signature of Risk Team Manager:

Acceptance

Assesment

Signature of Risk Project Manager:

Nature of Risk

Priority of Risk

Risk Level

Risk Remedy Action Team

Risk Remedy Wing Coordinator

Signature of Risk Manager:

Fig. 13: Risk assessment document

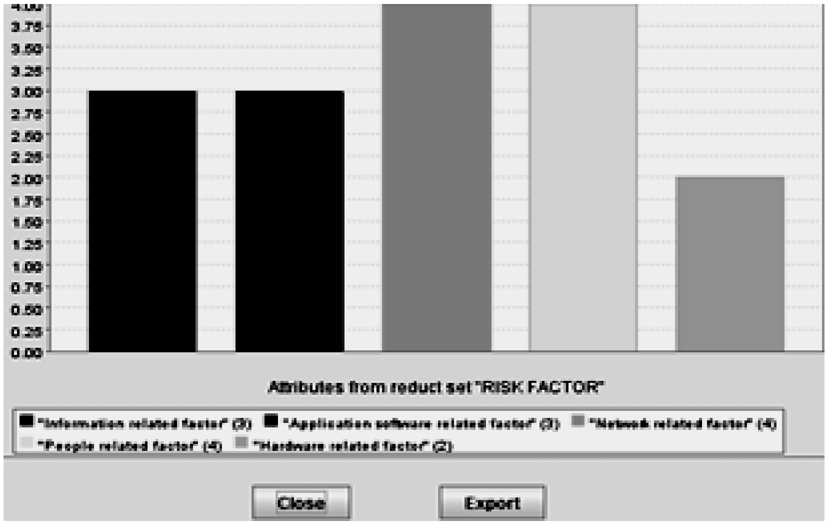

4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Attributes from reduct set "RISK FACTOR"
■ "Information related factor" (3)
■ "Application software related factor" (3)
■ "Network related factor" (4)
■ "People related factor" (4)
■ "Hardware related factor" (2)
Close
Export

Fig. 14: Result of attributes from reduced set

CONCLUSION

The earlier software security models including stride and octave security risk models focus on security goals of the software processes, activities in the organization not focusing on the external and internal various causes of that risk. In business and statistical computing, the different forms of noncompliance, causality and composability of multiple software components are considered to be the main causes of the security flaws that create different forms of risks in the application or the deployment or the overall system. In the proposed, six cause model, as per the risk pyramid layers, risk acceptance is expressed as a function through the comparison of the risk potential actuals with the past hacker events. The difference between expected and actual in risk potential is processed by the proposed holding function and a risk matrix has been constructed to identify the severity of the event and probability of occurrence of the event.

In the distributed environment, a risk assessment document will be automatically generated and forwarded to the nodes in the network. The accepted risk is assessed using fuzzy rough set model since the uncertainties of attacks are fuzzy in nature with minimum and maximum permitted ranges. The risk factor is fuzzified based on security attack activity with respect to impact on information content loss in percentage, impact on the application of software service loss, impact on the network authenticity loss, impact on people or social networking, impact on hardware resources loss and impact based on hacker activity using rough set exploration system tool. The process risks that lead to complexity and controllability factors of the vulnerable features of the system are monitored and documented so as to minimize the risks across the people during the evolution of any product. In the context of software security, the risk analysis has been carried out in the design stages focusing on the network, hardware software, hacker related features to accept and then assess the risk.

The research work inferred that in fuzzy rough set approach, the decomposition tree is analysed with various nodes whereas if the total number of nodes in a tree is 21 and level of decomposition is 4 there is no rule match for risk analysis with respect to decomposition of attack complexity. When the complexity factor of the secured component is 0.7 the risk factor increased to 0.8 whereas if the compliance factor of the software module is 0.4 the risk factor is increased to 0.8, if the criticality factor of the secured modules is 0.8 the risk factor increased to 0.64, if the composability factor of security module is 0.6 the risk factor increased to 0.8, the controllability of secured module is increased with risk factor of 0.8 and if the confidentiality of secured component is 0.7 the risk factor increased to 0.8. The risk analysis is done and the risk factor is found to be increasing to a maximum 0.64 and decreasing to a minimum of 0.1 where the factors are scaled up to a value 1 for all types of attacks. The series limitation in the proposed research work is the scaling of attacks and their impacts with respect to each organization is neglected and assumed to be constant. If the impact and the contribution of the one factor are dependent on the other parameter, then the six cause models of software security cannot be entertained to reason out the attack. The calibration segment of the various factors in the software security is also another limitation based on which the impact of the application software cannot be accurately quantified. The fuzzy techniques are helpful in overcoming the said limitations and achieve the best possible reasoning in the software security processes. The research contributes two important functional approaches as acceptance function and assessment functions to quantify the risks of the software while being deployed as an application in a networked scenario. The maximum risk factors in the software security model as per the proposed limited six cause factors are studied quantitatiely with respect to compliance and complexity and so on, the important aspect of today software business and health care application.

REFERENCES

Gilliam, D.P., 2004. Security risks: Management and mitigation in the software life cycle. Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June 14-16, 2004, IEEE, USA., ISBN: 0-7695-2183-5, pp: 211-216.

HajSaid, F., Y. Hassouneh and H. Ammar, 2011. Security risk assessment of software architecture. Proceedings of the 21st International Conference on Computer Theory and Applications (ICCTA 2011), ICCTA, Alexandria, Egypt, October 15-17, 2011, pp: 15-17.

Hamdaqa, M. and A.L. Hamou, 2011. An approach based on citation analysis to support effective handling of regulatory compliance. Future Gener. Comput. Syst., 27: 395-410.

Jensen, R. and Q. Shen, 2004. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE Trans. Knowledge Data Eng., 16: 1457-1471.

Jensen, R. and Q. Shen, 2007. Fuzzy-rough sets assisted attribute selection. Fuzzy Syst. IEEE. Trans., 15: 73-89.

Lee, S.W., 2011. Probabilistic risk assessment for security requirements: A preliminary study. Proceedings of the 5th International IEEE. Conference on Secure Software Integration and Reliability Improvement, June 27-29, 2011, IEEE, Jeju Island, ISBN: 978-1-4577-0780-3, pp: 11-20.

McManus, J., 2004. Risk Management in Software Development Projects. ELSEVIER, Amsterdam, Netherlands, ISBN: 0-7506-5867-3, Pages: 169.

Mkpong, I.R., D. Umphress, J. Hamilton and J. Gilbert, 2007. Quantitative software security risk assessment model. Proceedings of the 2007 ACM Workshop on Quality of Protection, October 29-November 2, 2007, ACM, New York, USA., ISBN: 978-1-59593-885-5, pp: 31-33.

Ni, M., J.D. McCalley, V. Vittal, S. Greene, C.W.T. Ganugula and T. Tayyib, 2003. Software implementation of online risk-based security assessment. IEEE Trans. Power Syst., 18: 1164-1172.

Pattanaik, B. and C. Subramaniam, 2011. Fault detection in embedded system using rough and fuzzy rough sets. Proceedings of the 15th WSEAS International Conference on Computers, July 14-16, 2011, WSEAS, Wisconsin, USA., ISBN: 978-1-61804-019-0, pp: 405-411.

Pearl, J., 2009. Causality Models, Reasoning and Inference. 2nd Edn., Cambridge University Press, Cambridge, UK., ISBN: 9780521749190, .

Pistoia, M., N. Nagaratnam, L. Koved and A. Nadalin, 2004. Enterprise Java Security: Building secure J2EE Applications. Addison-Wesley Professional, Boston, USA., ISBN: 0-321-11889-8, Pages: 565.

Rehman, S. and K. Mustafa, 2009. Research on software design level security vulnerabilities. ACM. SIGSOFT. Software Eng. Notes, 34: 1-5.

Russell, R., 2000. Hack Proofing Your Network. 2nd Edn., Syngress, Maryland, USA., ISBN: 1-928994-15-6, Pages: 427.

Sharif, A.M. and S. Basri, 2011. A study on risk assessment for small and medium software development projects. Intl. J. New Comput. Archit. Appl., 1: 325-335.

Tevis, J.E.J. and J.A. Hamilton, 2004. Methods for the prevention, detection and removal of software security vulnerabilities. Proceedings of the 42nd Annual Southeast Regional Conference, April 2-3, 2004, Huntsville, AL., USA., pp: 197-202.

Yeung, D.S., D. Chen, E.C. Tsang, J.W. Lee and W. Xizhao, 2005. On the generalization of fuzzy rough sets. Fuzzy Syst. IEEE. Trans., 13: 343-361.

Zhang, Y.K., S.Y. Jiang, Y.A. Cui, B.W. Zhang and H. Xia, 2010. A qualitative and quantitative risk assessment method in software security. Proceedings of the 3rd Internationa IEEE. Conference on Advanced Computer Theory and Engineering, August 20-22, 2010, IEEE, Chengdu, China, ISBN: 978-1-4244-6539-2, pp: 534-539.