I.J. Modern Education and Computer Science, **2021, 6, 30-41**
Published Online December 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2021.06.04

Quantitative Analysis of Software Security through Fuzzy PROMETHEE-II Methodology: A Design Perspective

Suhel Ahmad Khan
Department of Computer Science, Indira Gandhi National Tribal University, Amarkantak, 484887, Madhya Pradesh, India
Email: ahmadsuhel28@gmail.com

Mohd Nadeem, Alka Agrawal, Raees Ahmad Khan
Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, 226025, Uttar Pradesh, India
Email: mohd.nadeem1155@gmail.com, alka_csjmu@yahoo.co.in, khanraees@yahoo.com

Rajeev Kumar
Department of Computer Science and Engineering, Babu Banarasi Das University, Lucknow, 226028, Uttar Pradesh, India
Email: rs0414@gmail.com

Received: 23 July 2020; Revised: 15 August 2020; Accepted: 18 November 2020; Published: 08 December 2021

Abstract: The objective of this research study is to develop secure and multi-functional software or web application with controlled complexity. The demand of software security in different IT sectors is the main focus of the present endeavor. The different design factors and their prioritization are the need and demand of the system. We have selected the case of banking software or application. Security assessment is an integral part of risk management practices which provides an analytical mechanism to control and integrate security features for valuable opinion during the design phase. The designing of secure software and the impact of security factor is adopted and evaluated by the Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE)-II method. The PROMETHEE-II methodology evaluates the impact of factors with respect to the design alternatives. The current priority is to work on the state-of-the-art security attributes or alternatives of software design. Decision makers are generally responsible for evaluating various responses within their technical or scientific jurisdiction and rank them accordingly. Fuzzy set theories are the most appropriate tools to provide results for modeling qualitative information because of their ability to handle the impreciseness that is common in rating alternatives. The proposed work highlights the effectiveness of fuzzy PROMETHEE-II method in this context. We have enlisted this methodology for comparing software security factors in design perspective by using linguistic variables. The quantitative analysis attempted in our study was highly accurate for evaluating the security attributes and ranking them as per their priority, particularly in the context of banking software design. The study concludes with the advantages of employing the Fuzzy PROMETHEE-II vis-à-vis the other methodologies in analyzing the software security in the context of design.

Index Terms: Software Security, Software Design Attributes, Fuzzification, Fuzzy PROMETHEE-II.

1. Introduction

Software and web application security in design phase require consistent improvisation. Security practitioners are continuously working on inventing mechanisms that enable efficacious use of software. As the instances of cyber theft rise, there is a compelling need to prioritize the design attributes or alternative as per the requirements of the software used in a particular field. The recent security trends in software design are primary concern in our research study. Software and web application security is involved in procedure of designing, testing and building, where the software identifies problem in itself.

The IT firms are not fully aware of the importance of security because of lack of knowledge of security threat in software design. Web application and software security is an emergent field of research and development. Nowadays the business models are fully based on software or web application. The recent information of security threats are SQL injection, command injection, buffer overrun and stack buffer overflow, etc. Some of the most alarming cases in this league are: *Sony Pictures* faced SQL injection in 2011 by Lulzsec (hacker group) due to which the data of one million users was released; the hackers breached the data of two million credit card users of *Citigroup* in 2020; *Apple* and *Uber* have also faced the same kind of security theft. In 2017, the *HBO*also lost its confidential information to the hackers' invasion [1].

The rampant increase in such instances has propelled NASCIO (the National Association of State Chief Information Officers) to conduct surveys in 2020 and revise its policy against the data collection of users by the organizations. One of the revisions as per the CCPA 2020 (California Consumer Privacy Act) states that the businesses must report the collected consumer data [2]. The advancement of AI (Artificial Intelligence), 5G and quantum computing also affect the software and web application security. According to CERT analysts at Carnegie Mellon University, "the most successful attacks result from targeting and exploiting known, unpatched software vulnerabilities and insecure software configurations; a significant number of which are introduced during software design and development [3]."

The major objectives of this research are: Analyzing the design attributes with security factors; determining the design weights and the ranks of the chosen factors by employing the methodology of Fuzzy PROMETHEE II in the web application or software development; conduct a comparison between different multi criteria decision making for emphasizing the effectiveness of the latter technique over the others, thereby establishing that the proposed methodology would ensure better design perspective with different security factors. The design metric factors of software are confidentiality, integrity, availability, authorization and authentication. The ranking of the design factors are evaluated by the selected PROMETHEE-II methodology. It is necessary to select the priority of each factor in the design and development of the software. We selected the PROMETHEE-II methodology over the Analytical Hierarchy Process (AHP) due to the presence of appropriate structural system [4, 28]. The PROMETHEE-II methodology gives consistent result. The PROMETHEE method has different versions. For this study, in particular, we have selected PROMETHEE-II methodology for quantitative research. The other versions of ranking methodology have different pros and cons. But with PROMETHEE-II, we were able to determine the complete ranking of the alternatives.

The rest of this study is organized as follows: Section 2 tabulates the relevant literature review, highlighting the main studies done in this domain. In section 3, different factors and alternatives of the software have been described. Methodology of PROMETHEE-II has been explained in section 4. Section 5 illustrates the case study of banking software which we have alluded to for empirical analysis. Section 6 presents the comparisons of different methodologies. The study concludes with key suggestions cited by the researchers in section 7.

2. Related Work

Several research pursuits have reckoned in the ranking of the design factors and alternatives of software design. After an extensive literature review of these research endeavors, we premised our study on the tenets of the following studies:

V. Balali et al. [2014] based their research on selection of methodology in ranking of the affecting factors by giving the idea of comparison between two AHP and PROMETHEE methodology [4]. They worked on building structural system and selected the ranking by using the led system of selection.

P. Murasanto et al. [2011] gave the rank of software by using different metrics. They gave the rank by object oriented codes. While AHP provided the weight of the software quality, the PROMETHEE methodology gave the rank of the quality of the software. The rank of the software was determined by the PROMETHEE [5]. The rank is beneficial for the user, have object oriented software design. The structure of software can be improved further to achieve better characteristics and better design of software.

S. K. Pandey et al. [2010] enlisted the vulnerabilities that are exploited by the attackers for their attack on software [6]. The research concluded that the present securities under attack like the firewall, intrusion detection and antivirus are not sufficient. The combined effort by the software development community would help in minimizing the attack on software. The study emphasized that security should be assured at each phase of software development.

G. McGraw's [2004] article is on exploring the software security [7].The article cites that software bugs like the design flaws like the error handling are the susceptible parts of software design. Malicious codes can hack system through these software defects. The up gradation and data addition can also be a security risk factor for the software.

K. Kaur et al. [2014] gave the existing component of the software a form of ranking. The study discusses components of the software and its requirement for the development [8]. The authors integrated the components of the software and evaluated the ranking of the software by the adopted methodology PROMETHEE. The required components of the software system were selected by the multi criteria decision making tool of the PROMETHEE. The PROMETHEE method gives the ranking of the components of the software.

S. Vinodh et al. [2012] gives the sustainability concept of manufacturing organization [9]. The used methodology of PROMETHEE for ranking the manufacturing organization. They classified the sustainability in economic, environmental and social perspective. The selection process includes the multi criteria decision making of the selection problems. The study gives the change of material for best orientation of the result.

V. Rao et al. [2010] have worked on different criteria used in manufacturing sector into ranking form and the weight of the criteria is evaluated by the fuzzy AHP [10]. The multi criteria decision making method used here is PROMETHEE. It can evaluate the weight of the criteria and rank the desired criteria for better applicability in the manufacturing sector.

The research studies cited above helped us to opt for multi criteria decision making for prioritizing the factors and different alternative of secure software in design perspective.

3. Factors and Alternatives of Software & Web Application

Security engineering requires exploration of new possibilities to trace vulnerable fragments at the time of design and development and build a system that is more resistible, resilience and tolerant against problems. The goals of software security assessment are to build defect-free software and limit the damages through reduced occurrence rate of failures. It is highly important to focus more on the most appropriate properties during software development. Latest techniques are more helpful in solving the conflicts among the selection of alternatives for the user's satisfaction. The intent of prioritization is to expose security-related defects, or any critical flaws leading to denial of service, degradation of service, or other undesired behavior at the time of software design phase. Our research paper is based on different security factors of software and web applications. Figure1 shows the different attributes of software security and Figure 2 shows the design factors. The different security factors or alternatives of the software security and web application security are discussed below:

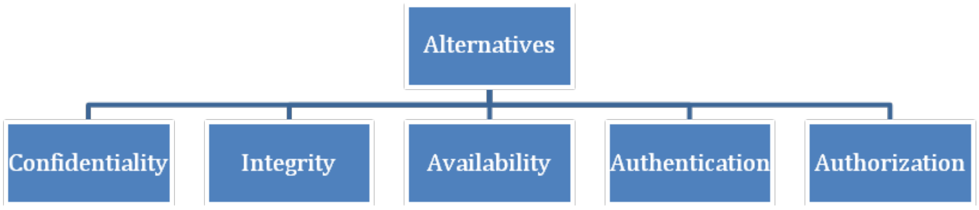

Fig. 1. Security Alternatives of Software and Web Application

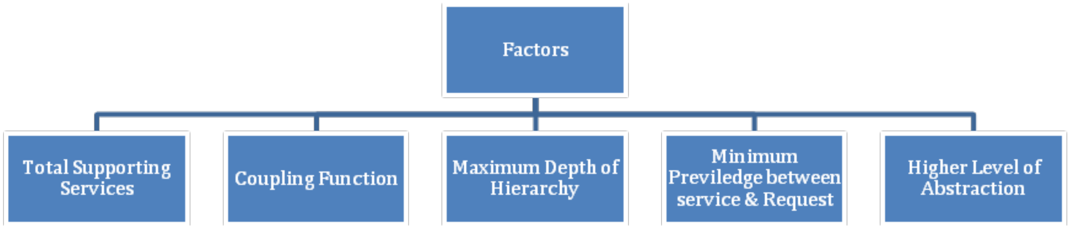
Factors
Total Supporting Services
Coupling Function
Maximum Depth of Hierarchy
Minimum Previledge between service & Request
Higher Level of Abstraction

Fig. 2. Design Factors

Confidentiality [C1], the most privileged and important factor of software security is confidentiality [11]. The confidentiality of the software must be secured from the malicious attacks. The organization has the necessary task to secure the confidentiality of the software. The data security is the important and primary consideration for the developers. Any breach can severely affect the confidentiality of software data, resulting in huge financial loss. For instance, the HBO lost millions of dollars due to phishing attack on the data containing the details of its subscription members.

Integrity [C2] is the key attribute of software security. The information of data is continuously changing in dynamic web application and software. The bank and e-commerce web sites updated their application nearly every day as per their clients' requirements [12]. The software integrity involves and authorizes the client's approval mechanism for the up gradation. The malware causes the interruption which may affect the undesirable modification and violate the web application and software integrity.

Availability [C3] is a necessary part of software and web application; without the attribute of availability the factors of software are pointless [11]. The client or user needs to access the software remotely so the availability is important. It is pointless if the information is not accessible, the unavailability is damaging factor for the organization.

Authentication [C4] is the process of identifying the user's identity to access the software [13]. Authentication depends on one or more authentication factors [14]. This is similar to authentication checks done for accessing email. The user must have both the authentication factors of id and password to access the email. The authentication factor is a very important security attribute.

Authorization [C5] has different phases in software and web application [15]. The developer makes the condition for the authorization phase of the software. The authorization has more than one step for the secure software and web application. It is the access control of the software. The access control is well authenticated by the developer to make the software secure.

Software Design Factors

The selection of design factor depends on the factors that would satisfy the required functionality of the software security. The selection of component or factor is also the major issue of software design. Here we have mention the design factor of software. The evaluations of affecting factor of attribute with respect to the factors are analyzed by the PROMETHEE method.

Total Supporting Services [CR1], to avoid the integrity violation in the design phase, total supporting services are used in the design phase of the software [16]. In the design phase, the entities are connected to each other by coupling and inheritance. Secure design time allows proper authorization and authentication. It is the design factor for the present research.

Coupling Function [CR2]-Coupling of the software system is the connection between the inter dispensaries of the function [17]. In the design phase of the software, the coupling function is the function in which the two or more functions are connected individually. It is a concept of coupling, inheritance, encapsulation of classes [16]. The design phase of the software has the coupling function which is a connection between the two phases of the design.

Maximum Depth of Hierarchy [CR3]-Depth hierarchy is the object orientation of design phase. It is the generalize hierarchy of the design phase [18]. Depth of inheritance is similar to fan down. Depth hierarchy of the design factor is defined at the system level. The maximum hierarchy of depth is the factor of software design.

Minimum Privilege between Services & Request [CR4] is the way to provide the low performance and more error condition in authorization attribute of the software [19]. It is the design factor of the following attributes. The effect of attribute on the following factors can be evaluated by the ranking methodology. In authorization, it provides the maximum strength of security of the software.

Higher level of Abstraction [CR5] has an important role in software reuse [20]. Concise and expressive abstractions are necessary. It can reduce the effort to regain the software [21]. The higher level of abstraction is the essential procedure of software design. The design factors of the software abstraction are important.

4. Methodology

The mathematical ranking methodology of PROMETHEE-II provides complete quantitative approach of the ranking [22]. The optimization problems of software are resolved by ranking of factors affecting the software on its priority. The problem of selection of alternatives is resolved by applying the Multi Criteria Decision Making (MCDM) approach [27]. This is considered for the evaluation of ranking.$C1, C2, .. C5$ Are the alternatives and $CR1, CR2, .. CR5$ are the design factors or criteria. MCDM approach is developed by this PROMETHEE-II method. MCDM will be analyzed and understandable with deeply required incomparability of the PROMETHEE-II [23]. The problem of MCDM is reduced by the criteria of optimize solution by the utility function. The utility function is exaggerated by the completely transforming the structure of MCDM. The paper [24] proposed the ranking relation with the dominance. The methodology of PROMETHEE-II methods are based on ranking relation. It does not give uniform alternative for each criterion. The inclination structure of PROMETHEE-II method is principally based on pair-wise individual comparison of each alternative. In methodology of PROMETHEE-II, the differences between the two alternatives of the criterion are considered. According to the deviation, the developer selects the preferences. The developer allocates best alternative for the small preference, and no preference for negligible deviation. The deviation and preference are directly proportional to the preferences. The preferences are the real numbers between 0 and 1. We fuzzified the MCDM approach of PROMETHEE-II, to evaluate the more appropriate value between 0 and 1.

Decision matrix is used for the best selection of alternative from the given alternatives. It is the evaluation tool for ranking the performance of each criterion. In the PROMETHEE-II technique, various decision support systems have the efficiency to calculate and analyze the alternatives having different conditions or the same condition. The developer gives the result of the linguistic queries raised by the client. The developer answers the query with quantitative and qualitative performance for the individual criterion. The determinations of alternatives with respect to the criteria are based on the understanding of the human decision and performance.

PROMETHEE-II Method

The PROMETHEE-II technique has the following seven steps:

Step-1 Space of the Problem

Decision matrix of C alternative and CR criteria are chosen to evaluate the dimension of the problem. The weight and type of each criterion is chosen.

Step-2 Normalization by minimum-maximum method

Normalize the Decision Matrix. This normalization can be done on the basis of direct and indirect criteria.

$$Minimum\ Rij = \frac{[Xij-(Xij)]}{[(Xij)-(Xij)]} \tag{1}$$

$$Maximum\ Rij = \frac{[(Xij)-Xij]}{[(Xij)-(Xij)]} \tag{2}$$

Where i= 1, 2….m & j=1, 2…..n

Step-3 Pair-wise comparison

Calculate the evaluation differences d_j of the alternatives $i^{th}$with respect to other alternatives for each criterion. The pair-wise comparison of the alternative calculated, here

$$d_j(a,b) = g_j(a) - g_j(b) \tag{3}$$

Step-4 Preference value

The PROMETHEE method has six preference functions. Here, we have used the usual preference function. This method cannot increase complexity and is always assigned 1, if$d > 0$, then the value after the comparison in d is lost.

$$P(d) = \{0\ d \leq 0d\ d > 0 \tag{4}$$

Step-5 Calculate the aggregated preference value

The preference aggregated value are evaluated by the equation 5, under condition that the weights of w_j satisfy.

$$\{\pi(a,b) = \sum_{j=1}^{k} \quad P_j(a,b)w_j\ \pi(b,a) = \sum_{j=1}^{k} \quad P_j(b,a)w_j \tag{5}$$

$$\sum_{j=1}^{k} \quad w_j = 1$$

Step-6: The leaving and entering outranking flow

Each alternative is related to the other alternative by calculating outranking flows. Leaving outranking (positive) flow for the a^{th} alternatives

$$\phi^{+}(a) = \frac{1}{n-1}\sum_{x\epsilon A} \quad \Pi\ (a,x) \tag{6}$$

Entering (negative) flow for the a^{th} alternatives

$$\phi^{-}(a) = \frac{1}{n-1}\sum_{x\epsilon A} \quad \Pi\ (x,a) \tag{7}$$

Step-7 Calculate the net outranking flow for the each alternative

$$\varphi(a) = \varphi^{+}(a) - \varphi^{-}(a) \tag{8}$$

Fuzzification

The PROMETHEE-II method lacks the ability to handle actual human decision environment data. Fuzzy concept is used to solve the uncertainties that arise due to human judgment. The Triangular Fuzzy Number (TFN) is based on the set of linguistics. So, the numbers on the intensity level of interest in AHP are transformed into the set of scale TFN. In the table 1 below, TFN is denoted as (l,m,u). Specialists designated marks to the elements influencing the qualities in a computable manner as indicated by scale that is exhibited in table 1. We have mentioned the values of TFN through the relation of factors with security of software or web application in design perspective.

Table 1. Triangular fuzzy scale

Intensity Values AHP	Definition	Triangular Fuzzy Scale
1	Very low	1, 1, 1
3	low	2,3, 4
5	Average	4 ,5, 6
7	High	6,7, 8
9	Very High	9, 9, 9
2	Weak	1,2, 3
4	Moderate Plus	3,4, 5
6	Strong Plus	5,6,7
8	Very, Very Strong	7,8,9

The procedure of weighting, which were given by researcher Buckley are mentioned in the following steps.

Step1: Table 1 is used for pair-wise comparison matrix to Triangular Fuzzy (TF) matrix M.

Step 2: The average arithmetic d_{ij} is evaluated by the expression

$$d_{ij} = \frac{\sum_{K-1}^{K} \quad d_{ij}^{k}}{K} \tag{9}$$

Here d_{ij}^{k} shows the k^{th} decision makers weight over the i^{th} condition over j^{th}condition by fuzzy TF member and K represents the no. of decision maker.

Step 3: The fuzzy weight of each condition is obtained by the criteria of Geometric Mean (GM)

$$\check{r_i} = \left(\prod_{j-1}^{n} \quad d_{ij}\right)^{\frac{1}{n}} \tag{10}$$

Here $i = 1,2, \ldots \ldots n$

Here $\check{r_i}$ represents triangular values, $\left(\prod_{j-1}^{n} \quad d_{ij}\right)^{\frac{1}{n}}$ represents GM of fuzzy comparison with each condition and n means the number of decisions. The TFN is obtained by the condition

$$w_i = \check{r_i} \otimes \ \left(\check{r_1} \oplus \ \check{r_2} \oplus \ldots.. \oplus \check{r_n}\right)^{-1} \tag{11}$$
$$= (lw_i, mw_i, uw_i)$$

Here w_i are the fuzzy criteria of weight; there are three types of arithmetic operation addition, multiplication and inverse in TFN. Since $r_1 = (l_1, m_1, u_1)$ and $r_2 = (l_2, m_2, u_2)$, then addition is: $r_1 \oplus \ r_2 = (l_1 + l_2,\ m_1 + m_2,\ u_1 + u_2)$multiplication: $r_1 \otimes \ r_2 = \ (l_1 . l_2,\ m_1 . m_2,\ u_1 . u_2)$ and inverse: $r_1 = (l_1, m_1, u_1)^{-1}$.

Step 4: Defuzzification methods are applied after evaluating the weight criteria. Non fuzzy value of M is obtained by the value of fuzzy number w in the given condition,

$$M_i = \frac{(lw_i + mw_i + uw_i)}{3} \tag{12}$$

$M_i$is a non fuzzy number, lw$_i$ is lower weight, mw $_i$ medium weight and uw$_i$is the upper weight . After this, we obtained the normalized value of N_i.

Step 5: After getting each value ofN_i, global weighting of the criteria is obtained by multiplying the local weight with weight-related within the same dimension.

Fuzzy PROMETHEE-II Method

The Fuzzy PROMETHEE-II method is applied on selecting the alternatives of web or software based security factors. The methods are proposed to figure out the best alternatives among the software security proposed criteria by the fuzzy distance between alternatives. The PROMETHEE-II methodology was introduced by Bran's et al. [1]. To figure out the rank and most influential factor of web or software security at design phase. The MCDM technique to determine the various factors of software security in linguistic assessment has already been mentioned in the previous section. The assessment of weights for each criteria and preference function are the prime prerequisites of this evaluation technique [11]. These criteria are represented in the form of trapezoidal fuzzy numbers and evaluations of fuzzy distances are managed in PROMETHEE-II method. Fuzzy PROMETHEE-II methods have been outlined in the following steps:

Step-1 Decision Making: First step is decision making in which the$K$ decision makers are represented by trapezoidal fuzzy numbers with membership function as shown in table 2.

Step-2 Evaluates Criteria: Second step involves generating feasible alternatives and factors. Evaluate the criteria with respect to the alternative. Here C is alternative and CR is criteria, as shown in table 5.

Step-3 Selecting the appropriate variables and their respective TFNs: They are used for evaluating the weights of criteria and the ratings of alternative under various criteria in table 6.

Step-4The $\boldsymbol{K^{th}}$decision makers in the form of TFN give the fuzzy rating: After this the fuzzy weight of each criterion is aggregated. The normalized fuzzy decision matrix is formed by maximum and minimum normalization method.

Step-5 Normalized decision matrix is weighted by multiplying the weights of evaluation criteria and the values in the normalized fuzzy decision matrix. This is known as the weighted normalized fuzzy decision matrix.

Step-6In this step, we compared the alternative in each criterion. After comparing, we calculated the maximum upper bound and distance of criteria. Preference function is calculated in table 7.

Step-7Fuzzy preference function is calculated to determine the value of the outranking relation.

Step-8 The leaving and outranking flows are calculated for ranking of alternatives as shown in table 8 and 9.

Step-9The net flow of preference is calculated and tabulated in table 10.

Step-10The final step is evaluating the preference ranking by constructing a value out braking graph.

5. Numerical Analysis

We have selected the case of banking system, where security is the cardinal issue. We cannot imagine the banking portal or software without security. The present security modules used in banking are not sufficient. The upgradation of security is necessary. Here, the initial value of evaluation has been taken from the TFN from table 1. Thereafter, we have evaluated the weight of the given value. Table 2 shows the initial TFN value of software design alternatives from the equation 12.

Table 2. TFN Value of Alternatives and Factor

	CR1	CR2	CR3	CR4	CR5
C1	1	5	3	3	4
C2	0.2	1	0.33	0.33	3
C3	0.33	3	1	0.33	3
C4	0.33	3	3	1	3
C5	0.25	0.33	0.33	0.33	1

Table 3 depicts the average value of alternative, evaluated by the equation 1 for minimum value and equation 2 for maximum value. The last column of table 2 shows the average value of alternative.

Table 3. Average Value of Alternative

						Average Value
C1	0.473	0.405	0.391	0.601	0.285	0.432
C2	0.094	0.081	0.043	0.066	0.214	0.099
C3	0.156	0.243	0.131	0.066	0.214	0.162
C4	0.156	0.0243	0.392	0.200	0.214	0.241
C5	0.118	0.026	0.043	0.066	0.071	0.065

Average values from table 3 are multiplied with the comparative values of alternatives in table 4 from the equation 5. We evaluated the preference value.

Table 4. Preference Value of Alternative

Weights	0.431	0.09	0.16	0.24	0.06
D(C1-C2)	1	0.85	1	1	0.34
D(C1-C3)	-0.83	0.43	0.75	1	0.34
D(C1-C4)	-0.83	0.43	0	-0.75	0.34
D(C1-C5)	-0.93	1	1	1	1
D(C2-C1)	1	-0.86	-1	-1	-0.34
D(C2-C3)	0.17	-0.42	-0.25	0	0
D(C2-C4)	0.17	-0.42	-1	-0.25	0
D(C2-C5)	0.07	0.143	0	0	0.66
D(C3-C1)	0.833	0.57	-0.75	-1	-0.34
D(C3-C2)	-0.16	0.43	0.25	0	0
D(C3-C4)	0	0	-0.75	0.25	0
D(C3-C5)	-0.09	0.57	-0.25	0	0.66
D(C4-C1)	0.833	-0.43	0	-0.75	-0.34
D(C4-C2)	-0.16	0.43	1	.025	0
D(C4-C3)	0	0	0.75	0.25	0
D(C4-C5)	-0.09	0.57	1	0.25	0.66
D(C5-C1)	0.93	-1	-1	-1	-1
D(C5-C2)	-0.07	-0.86	0	0	-0.66
D(C5-C3)	0.1	-0.57	-0.25	0	-0.66
D(C5-C4)	0.1	-0.57	-1	-0.25	-0.66

The updated table is shown below; the evaluation was done as per the Step 7 of fuzzy PROMETHEE and equation 5. The evaluation is depicted in table 5.

Table 5. Aggregated Preference Value

D(C1-C2)	0.43	0.07	0.16	0.24	0.02	0.92
D(C1-C3)	0	0.03	0.12	.024	0.02	0.41
D(C1-C4)	0	0.03	0	0	0.02	0.05
D(C1-C5)	0	0.09	0.16	.024	0.06	.055
D(C2-C1)	0.43	0	0	0	0	.043
D(C2-C3)	0.07	0	0	0	0	0.07
D(C2-C4)	0.07	0	0	0	0	0.07
D(C2-C5)	0.03	0.01	0	0	0.22	0.26
D(C3-C1)	0.39	0.05	0	0	0	0.44
D(C3-C2)	0	0.03	0.04	0	0	0.07
D(C3-C4)	0	0	0	0	0	0
D(C3-C5)	0	0.05	0	0	0.22	0.27
D(C4-C1)	0.39	0	0	0	0	.039
D(C4-C2)	0	0.03	0.16	0.06	0	0.25
D(C4-C3)	0	0	0.12	0.06	0	0.18
D(C4-C5)	0	0.05	0.16	0.06	0.22	0.49
D(C5-C1)	0.39	0	0	0	0	0.39
D(C5-C2)	0	0	0	0	0	0
D(C5-C3)	0.04	0	0	0	0	0.04
D(C5-C4)	0.04	0	0	0	0	0.04

The alternatives have a relation with alternatives, so we evaluated the outranking flow for the a^{th} alternative as shown in table 6. This was determined by equation 6 and 7.

Table 6. Leaving and entering the Outranking Flow

	CR1	CR2	CR3	CR4	CR5	φ +
C1	-	0.92	0.41	0.05	0.55	0.483
C2	0.43	-	0.07	0.07	0.26	0.207
C3	0.44	0.07	-	0	0.07	0.195
C4	0.39	0.25	0.18	-	0.49	0.327
C5	0.39	0	0.04	0.04	-	0.117
φ -	0.412	0.31	0.175	0.04	0.392	-

The ranking of alternative form equation 8 is shown in table 7.

Table 7. Outranking flow of alternative

	φ +	φ -	φ (Total)	Rank
Confidentiality 1	0.483	0.412	0.071	2
Integrity 2	0.207	0.31	0.032	3
Availability 3	0.195	0.175	0.02	4
Authentication 4	0.327	0.04	0.287	1
Authorization 5	0.117	0.392	-0.275	5

The ranking table shows the methodology of PROMETHEE used to evaluate the value and calculate the authentication alternative of the software.

6. Comparison of Different Methodologies

In table-8, we have mentioned the value of different ranking methods. The comparison shows that PROMETHEE-II methodology gives the appropriate and exact evaluations compared to the other ranking methodologies.

Table 8- comparison of different ranking methods

	Fuzzy PROMETHEE-II	Fuzzy PROMETHEE-I	Fuzzy VIKOR	Fuzzy ELECTRE	Fuzzy TOPSIS
Confidentiality	0.071	0.880	4.372	4.029	0.622
Integrity	0.032	-0.199	0.369	4.648	0.587
Availability	0.020	2.359	-4.711	5.160	0.515
Authentication	0.287	2.681	1.114	1.300	0.695
Authorization	-0.275	-0.100	-5.179	5.703	0.495
		R=0.6928	R=0.6375	R= 0.8850	R=0.8879

R= Co-relation (Pearson)

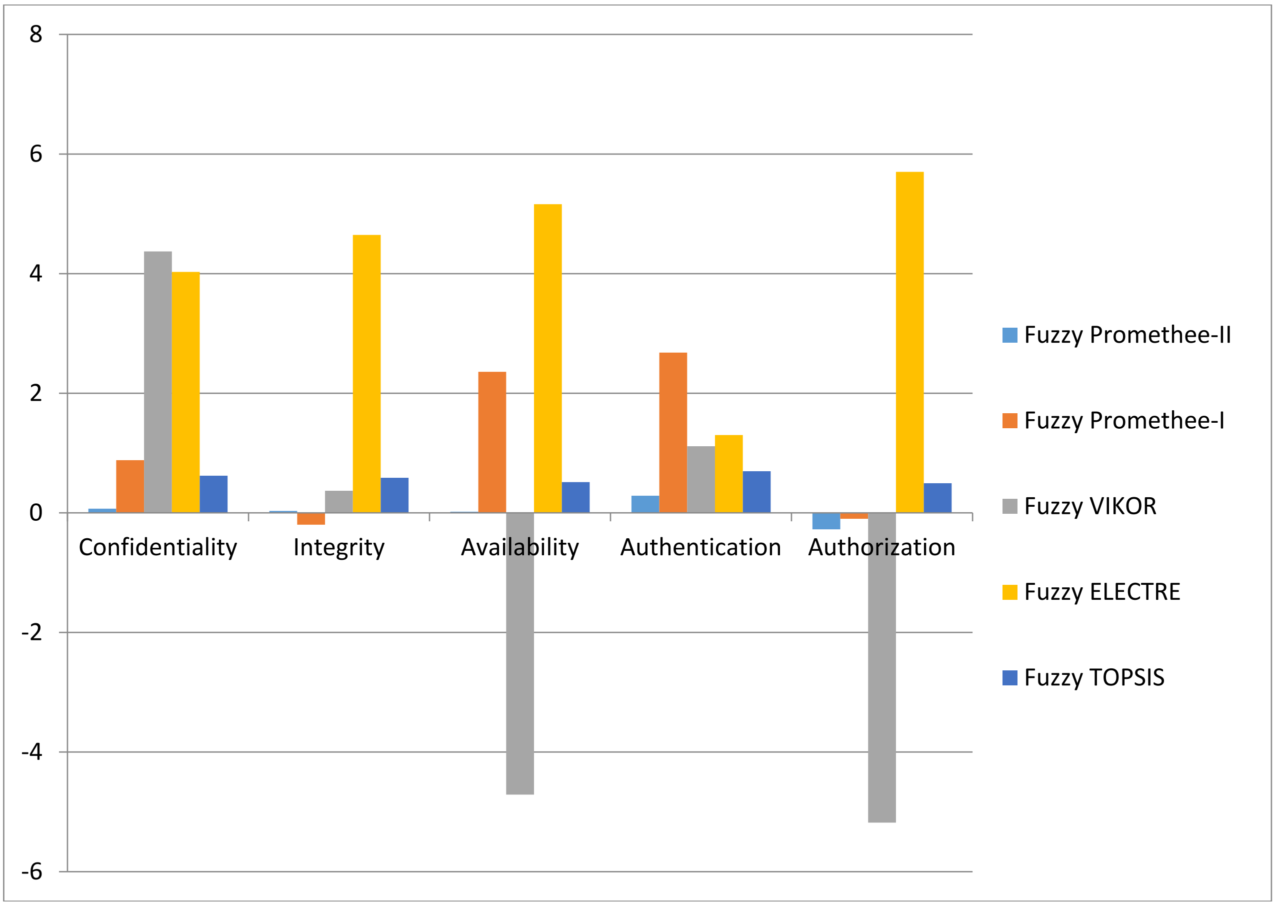

Fig. 3. Comparison between different preference methods

The different methods of preference comparison are shown in figure 3. The fuzzy PROMETHEE-II methodology gives the exact evaluated value compared to the other methodology. The developer prefers the alternative of authentication in the priority to develop the secure software.

7. Recommendations

Based on the empirical findings of our study, we have outlined four key suggestions that can be incorporated for the evaluation of the ranking of attributes to enhance the software security in design perspective. These are:

Suggestion 1: Ensure the use Unique Identification or access code (User id, password), transaction id, and encrypted information as method or attributes for each user.

The need for this is evident as is shown through the case study of internet banking. This contains the unique information for identification like user id, password, transaction id are {Customer, transaction, account details, web services}. There is possibility that an attacker can compromise the vulnerable attributes and perform illegal operations.

Suggestion 2: Crosscheck if the services from the other classes are sharing relevant credential information through inheritance/aggregation from base class to derived class.

Attackers can take advantage of such vulnerable methods that are being used to provide the services for user.

Suggestion 3: Generate a valid token with the corresponding unique identification or access code and validation through its corresponding access code.

An ATM card is being used as communication interface between the user and ATM via validating authenticity of user and his account no. The card holds sensitive information like {card id, pin} that the attackers can breach.

Suggestion 4: Examine the design; whether the design contains single/multiple/split/ multilevel password authentication system corresponding access code.

This will help in gauging how strong the authentication mechanism in the design is. The multiple authentication system is highly desirable to prevent fake or unauthenticated users.

On the basis of the above details, researchers traced the involved authentication classes manually and cross-examined them for vulnerable attributes or methods in design. It is mandatory to reduce the number of classes involved in authentication process for secure design. More number of classes provides higher attack surface for intruders to violate security through exploitable methods or attributes. The case study of internet banking reveals that during transactions, pin & transaction id can be the most vulnerable attributes [1, 2]. The intruders can violate the authenticity via these weak attributes. The most secure approach in this context would be to hide these attributes of classes that contain vulnerable attributes or methods. Once the user is validated as valid user, he can access the resources and these attributes can work in background to provide information from server to user channel. The reduced number of exploitable class increases the security of software design as authentication perspective.

8. Conclusion

The findings of the present study corroborate that the Fuzzy PROMETHEE II technique for security estimation on the basis of design perspective is the most significant procedure for verifying security estimation of web application. As explained in the introduction section of this study, an accurate and conclusive mechanism to evaluate the most important security factors would be a major contribution in developing secure structure of web application in the design perspective. Furthermore, evaluation and estimation are the best ways to accomplish Security Estimation of Web Application the board framework. This paper incorporates security factors and assesses these factors in a systematic framework. The results of this research investigation will assist the engineers in integrating the online secure administration framework along with structuring Security Estimation of Web Application with the reference of design tactics during its improvement.

Numerous estimation models in design tactics perspective or strategies have already been propositioned in several research resources for evaluating security independently. However, the accessibility of models or techniques which coordinate security on PROMETHEE strategy is altogether less. In this research paper, we reserved different alternatives of security risk of web based applications as per the opinions of the experts. These opinions necessarily centered on the contributing risk plan, mitigation plan and security attributes of the particular web based application. Information has been taken from the different websites. Results of the work will be an effective contribution in security estimation of web application system, helping the developers to devise more efficacious security risk mitigation strategies. Security estimation of web application in design perspective of the board framework is as yet overlooked.

This evaluation would assist the engineers in gaining knowledge about the structure of security. Recommendation can be delivered along with the suggested assessment procedure in this study so as to help the engineers in further refining structure of securities to suit the requirement of the systems. However, this recommendation may have a few limits of its own which can be addressed in future research initiatives. Limitation, as reckoned by the researchers of the present study, is that the information gathered for website architecture is noteworthy, but not encompassive. The results may contrast if the information is enormous. Moreover, there could be extra security configuration factors/attributes other than those that are recognized in this work.

The final ranking of the alternatives varies with normalized technique of PROMETHEE-II. The highest ranked alternative is authentication. The case study and importance of authentication adopted from [13], has the Authentication Quantification Model (AQM) hierarchy of banking system. It is the important evaluation of the authentication attribute with respect to the design factors. The evaluation value of the AQM explains the attributes of the authentication class with respect to the other attributes. The evaluation of AQM attribute is 0.445. The PROMETHEE-II methodology was used to determine the ranking of the attributes, which established the authentication mechanism. The selected case study of the banking system has costumers’ details of account, transaction and web services. The software attributes influence the authentication mechanism. Here, the 40% of the attributes have vulnerable property and need to be controlled for the strong mechanism of authentication for the security of software and web application.

Acknowledgement

This work will be acknowledged for the progress of the project number CST/D-2408 issued from UPCST.

References

[1] Cunningham, M. (2016). Complying with international data protection law. U. Cin. L. Rev., 84, 421.

[2] CCPA, D. U. (2020). California Consumer Privacy Act (CCPA) Website Policy. Policy.

[3] Chandler, J. A. (2003). Security in cyberspace: combatting distributed denial of service attacks. U. Ottawa L. & Tech. J., 1, 231.

[4] Balali, Vahid&Zahraie, Banafsheh&Roozbahani, Abbas. (2014). A Comparison of AHP and PROMETHEE Family Decision Making Methods for Selection of Building Structural System. American Journal of Civil Engineering and Architecture. 2. 149-159. 10.12691/ajcea-2-5-1.

[5] Mursanto, Petrus& Halim, Arwin. (2014). Combination of AHP and PROMETHEE for Measuring Quality of Object Oriented Software Design. 10.13033/isahp.y2014.055.

[6] Pandey, S. K., & Mustafa, K. (2010). Security Assurance: An Authentication Initiative by Checklists. International Journal of Advanced Research in Computer Science, 1(2).

[7] G. McGraw, "Software security," in IEEE Security & Privacy, vol. 2, no. 2, pp. 80-83, March-April 2004.

[8] Kaur, K., & Singh, H. (2014, May). PROMETHEE based component evaluation and selection for Component Based Software Engineering. In 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp. 1421-1425). IEEE.

[9] Vinodh, S., &Girubha, R. J. (2012). PROMETHEE based sustainable concept selection. Applied Mathematical Modelling, 36(11), 5301-5308.

[10] Venkata Rao, R., & Patel, B. K. (2010). Decision making in the manufacturing environment using an improved PROMETHEE method. International Journal of Production Research, 48(16), 4665-4682.

[11] Shakiba-Herfeh, M., Chorti, A., & Poor, H. V. (2020). Physical Layer Security: Authentication, Integrity and Confidentiality. arXiv preprint arXiv:2001.07153.

[12] Garg A., Mittal N., Diksha (2020) A Security and Confidentiality Survey in Wireless Internet of Things (IoT). In: Balas V., Solanki V., Kumar R. (eds) Internet of Things and Big Data Applications. Intelligent Systems Reference Library, vol 180. Springer, Cham

[13] Khan S. A. & Khan R. A. (2013), Security Quantification Model, , International Journal of Software Engineering IJSE, ISSN: 2090-1801, Volume 6, No. 2, 2013, pp: 75-89

[14] H. Hu and G. Ahn, "Constructing Authorization Systems Using Assurance Management Framework," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 4, pp. 396-405, July 2010.

[15] Khan, R. (2011). Secure software development: a prescriptive framework. Computer Fraud & Security, 2011(8), 12-20.

[16] Khan, S. A., & Khan, R. A. (2012). Integrity quantification model for object oriented design. ACM SIGSOFT Software Engineering Notes, 37(2), 1-3.

[17] Allen, E. B., Khoshgoftaar, T. M., & Chen, Y. (2001, April). Measuring coupling and cohesion of software modules: an information-theory approach. In Proceedings Seventh International Software Metrics Symposium (pp. 124-134). IEEE.

[18] Tegarden, D. P., Sheetz, S. D., &Monarchi, D. E. (1995). A software complexity model of object-oriented systems. Decision Support Systems, 13(3-4), 241-262.

[19] Khan, S. A., & Khan, R. A. (2010). Securing object oriented design: A complexity perspective. International Journal of Computer Applications, 8(13).

[20] Kang, N., Liu, Z., Rexford, J., & Walker, D. (2013, December). Optimizing the" one big switch" abstraction in software-defined networks. In Proceedings of the ninth ACM conference on Emerging networking experiments and technologies (pp. 13-24).

[21] Atheel K. Abdulzahra, Turki Y. Abdalla, (2019), Fuzzy Sliding Mode Control Scheme for Vehicle Active Suspension System Optimized by ABC Algorithm, International Journal of Intelligent Systems and Applications, Vol.11, No.12, pp.1-10.

[22] Brans, J. P., Vincke, P., &Mareschal, B. (1986). How to select and how to rank projects: The PROMETHEE method. European journal of operational research, 24(2), 228-238.

[23] Eppe, S., De Smet, Y., &Stützle, T. (2011, October). A bi-objective optimization model to eliciting decision maker’s preferences for the PROMETHEE II method. In International Conference on Algorithmic Decision Theory (pp. 56-66). Springer, Berlin, Heidelberg.

[24] Roy, B. (1977). Partial preference analysis and decision aid: The fuzzy outranking relation concept. Conflicting objectives in Decisions, 40-75.

[25] Kang, N., Liu, Z., Rexford, J., & Walker, D. (2013, December). Optimizing the" one big switch" abstraction in software-defined networks. In Proceedings of the ninth ACM conference on Emerging networking experiments and technologies (pp. 13-24).

[26] Charles W. Krueger. 1992. Software reuse. ACM Comput. Surv. 24, 2 (June 1992), 131–183. DOI:https://doi.org/10.1145/130844.130856.

[27] Maselle J. K., Mashaka J. M., Verdiana G. M， (2020), Multi-Criteria Decision Making and Numerical Optimization Approaches for Optimizing Water Loss Management Strategies in Water Distribution System - A case of Urban Water Supply and Sanitation Authorities in Tanzania , International Journal of Mathematical Sciences and Computing, Vol.6, No.1, pp.10-24, 2020.

[28] Farhad L., Kimia F., Nasrin B., (2020), An Analysis of Key Factors to Mobile Health Adoption using Fuzzy AHP, International Journal of Information Technology and Computer Science, Vol.12, No.2, pp.1-17.

Authors’ Profiles

Dr. Suhel Ahmad Khan is currently working as an Assistant Professor in the Department of Computer Science, Indira Gandhi National Tribal University (A Central University), Amarkantak, Madhya Pradesh. He has 10 year of teaching & research experience. His areas of interest are Software Engineering, Software Security, Security Testing, Cyber Security, and Network Security. He has completed one major research project with PI funded by UGC, New Delhi. He has published numerous papers in international journals and conferences including IEEE, Elsevier, IGI Global and Springer etc. Dr. Suhel Ahmad Khan is an active member of various professional bodies IAENG, ISOC-USA, IACSIT, and UACEE.

Mr. Mohd. Nadeem is currently working as a Full-Time Research Scholar at the Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), VidyaVihar, Raibareli Road, Lucknow, India and he has completed his M-Tech Degree in Electronic Circuit and System from Integral University, Dashauli, Lucknow, India in 2013. He has completed his B-Tech Degree in Electronics and Communication from Chandrasekhar Azad University Kanpur, India in 2010. His research interests are in the areas of Software Security, Quantum Security, Network Security, Security Risk, and IoT Security.

Dr. Rajeev Kumar completed the Master’s and PhD degrees in Information Technology from Babasaheb Bhimrao Ambedkar University, Lucknow, India, in 2014 and 2018, respectively. He has more than seven years of research and teaching experience. Dr. Kumar is currently working as an Associate Professor in the Department of Computer Science and Engineering, Babu Banarasi Das University, Lucknow, UP, India. His research interest includes different areas of Security Engineering.

Dr. Alka Agrawal is currently working as an Assistant Professor in the Department of Information Technology, Babasaheb Bhimrao Ambedkar University, (A Central University), VidyaVihar, Raibareli Road, Lucknow, India. Dr. Alka has more than 13 years of teaching & research experience and she has more than 100 research publications with good impact factors in reputed International Journals and Conferences including IEEE, Springer, Elsevier, Inderscience, Hindawi, and IGI Global etc. She has also authored National and International Books. Her research interests are in the different areas of Security Engineering and Computational Techniques.

Prof. Raees Ahmad Khan (Member, IEEE, ACM, CSI etc.) is currently working as a Professor & the Head of the Department in the Department of Information Technology, Dean of School for Information Science & Technology, Babasaheb Bhimrao Ambedkar University, (A Central University), VidyaVihar, Raibareli Road, Lucknow, India. Prof. Khan has more than 20 years of teaching & research experience and he has more than 300 research publications to his credit with good impact factors in reputed International Journals and Conferences including *IEEE, Springer, Elsevier, Inderscience, Hindawi, and IGI Global,* etc. He also has authored and edited a number of National and International Books in English and Chinese Language. His research interests are in different areas of Security Engineering and Computational Techniques.

How to cite this paper: Suhel Ahmad Khan, Mohd Nadeem, Alka Agrawal, Raees Ahmad Khan, Rajeev Kumar, " Quantitative Analysis of Software Security through Fuzzy PROMETHEE-II Methodology: A Design Perspective", International Journal of Modern Education and Computer Science(IJMECS), Vol.13, No.6, pp. 30-41, 2021.DOI: 10.5815/ijmecs.2021.06.04