ATLANTIS PRESS International Journal of Networked and Distributed Computing, Vol. 6, No. 1 (January 2018) 1–10

Assessing Risk of Security Non-compliance of Banking Security Requirements Based on Attack Patterns

Krissada Rongrat[1], Twittie Senivongse[2]

[1] *Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 254 Phyathai Road, Wangmai, Pathumwan, Bangkok, 10330, Thailand*
E-mail: krissada.ro@student.chula.ac.th

[2] *Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 254 Phyathai Road, Wangmai, Pathumwan Bangkok, 10330, Thailand*
E-mail: twittie.s@chula.ac.th

Abstract

Information systems such as those in the Banking sector need to comply with security regulations to assure that necessary security controls are in place. This paper presents an initial risk assessment method to assist a banking information system project in validating security requirements of the system. Dissimilarity between the textual security requirements of the system and the security regulations is determined to identify security non-compliance. A risk index model is then proposed to determine the risk level based on the severity and likelihood of exploit of any security attack patterns that could potentially affect the system if the missing regulations are not implemented. In an experiment using a case study of nine Thai commercial banks and the IT Best Practices of the Bank of Thailand as the regulations, the performance of compliance checking is evaluated in terms of F-measure and accuracy. It is also found that there is a strong positive correlation, with the coefficient of over 0.6, between the risk indices from the method and the security expert judgment.

Keywords: Security requirement, Risk assessment, Security attack pattern, Regulatory compliance, Text similarity, Banking.

1. Introduction

Security is one of the most crucial attributes that must be taken into account during the development of an information system. In both contexts of on-premise and cloud-based system solutions, necessary security controls or services have to be in place to safeguard critical information and business operations of the system.[1, 2] In the banking sector, there is an increase in the number, sophistication, and scope of cyber attacks against the industry.[3] It is essential that security concern needs to be part of the development of any banking information system from the beginning. Therefore, security requirements of the system must address security matters in a complete controlled structured way on the basis of recognized standards and best practices.

This paper uses the case of the banking sector in Thailand as a case study. Security requirements of an

information system of a commercial bank in Thailand have to comply with a set of regulations called IT Best Practices.[4, 5] The IT Best Practices is a regulatory agreement between the Bank of Thailand (BOT) and commercial banks in Thailand to ensure that, when any commercial bank needs to develop or customize an information system, security requirements of the system must be validated to check their compliance with the IT Best Practices early at the beginning of the project before proceeding to development. On the other hand, the bank can develop the system first, but before launching it to production, the system security requirements must be validated. In normal practice, before the validation by internal auditors of the bank and auditors from the BOT, security requirements are validated initially by either the requirements engineers or business analysts of the project. Since the validation requires a study of textual security requirements and IT Best Practices to determine if the regulations are met, this consumes project time and cost and largely relies on knowledge and experience of the requirements engineers and business analysts. Misjudgment could mean that some regulations are merely partially met or even missing and it would be more costly to find that out later in the project or leave it until the auditors find out.

This paper presents a method to help requirements engineers and business analysts of a banking system project to assess the security requirements of the system to be developed. The assessment comprises 1) checking compliance with the IT Best Practices and 2) assessment of risk associated with non-compliant requirements. On checking compliance, text similarity analysis is used to determine which security practices are missing from the security requirements document. Given those missing practices, we determine potential security attacks that could occur and assess the degree of risk based on the harm those attacks can do to the system. To assess the risk, we use the CAPEC attack pattern classification[6] to build a risk index model. The assessment result identifies non-compliant locations within the security requirements document and the degree of risk of potential attacks if the system is implemented based on such incomplete requirements. We evaluate the performance of compliance checking as well as validity of risk assessment.

Section 2 of this paper presents important background of the work. Section 3 discusses related research. Section 4 describes the proposed risk assessment method. An evaluation is shown in section 5 and the paper concludes in section 6.

2. Background

2.1. IT Best Practices

IT Best Practices[4, 5] is a document developed by the Bank of Thailand (BOT) and commercial banks in Thailand as a recommendation of security solutions to control risk that could occur to banking information systems. The recommendation is based on cybersecurity frameworks developed by NIST, ISO 27005, COBIT etc. and covers operation procedures, operation controls, and information systems. The risk control part of the IT Best Practices specifies baseline requirements for banking information systems and is the most relevant in the context of this paper. The baseline requirements address five domains of information systems: 1) Core Banking Application, 2) ATM Application Control, 3) ATM Machine, 4) Internet Banking Application Control, and 5) Internet Banking Security.

2.2. CAPEC Attack Pattern

Attack patterns document reusable attack knowledge to bridge the knowledge gap and assist with attack analysis.[7] The Common Attack Pattern Enumeration and Classification (CAPEC)[6] is a taxonomy of cyber security attacks developed by MITRE corporation. It has been incrementally built, starting from 2007, and includes a collection of attack patterns. Each attack pattern captures knowledge about how specific parts of an attack are designed and executed, and gives guidance on how to mitigate the effectiveness of the attack. Each attack pattern description includes several topics, e.g., Summary, Attack Execution Flow, Typical Severity, Typical Likelihood of Exploit, Methods of Attack, Examples-Instances, Attackers Skills and Knowledge Required, Resources Required, Solutions and Mitigations, Related Weaknesses, Relevant Security Requirements, Confidentiality/Integrity/Availability Impact, Technical Context. Among these, we use Solutions and Mitigations, Relevant Security Requirements, Typical Severity, and Typical Likelihood of Exploit information for risk assessment.

3. Related Work

In this section, we discuss related work on application of text analysis to software requirements and security risk assessment.

On application of text analysis to software specification, Stierna and Rowe[8] argue that finding opportunities for reuse of previously written software modules in large and complex systems is difficult. Instead, the reuse opportunities can be found indirectly through software requirements. That is, they match written requirements of the new software against the requirements used to define the old software, and requirement pairs with common words suggest reuse of software modules related to the old requirements. We follow their approach in processing textual software requirements and using Cosine coefficient[9] to measure the degree of similarity between requirements. Ilyas and Kung[10] present a requirement similarity measurement framework to support similarity measurement for the requirements of a running project and the requirements of the already completed projects. They use Dice, Jaccard, and Cosine coefficients as similarity measures. Once similar requirements are found, the design and code of the already completed projects become reusable components. Dag *et al.*[11] use an automated analysis of flow of software requirements to increase efficiency of their requirements engineering process. When there are new requirements coming continuously from many different sources and having to be responded quickly for short time-to-market, they need to identify relationships between requirements. They use text similarity analysis to find duplicate requirements so that they can avoid doing the same job twice, assigning the same requirement to different developers, or getting two solutions to the same problem. Also, they use Dice, Jaccard, and Cosine coefficients as similarity measures but Dice and Cosine coefficients perform better in the experiment.

On security risk assessment, Yu *et al.*[12] presents an automated tool to support the use of formal logic, i.e., security argumentation, to determine security satisfaction of security requirements, or arguments. The tool includes a Lucene-based search engine for security attacks and weaknesses information which is taken from CAPEC and CWE catalogs. Keywords from the arguments are searched for relevant attacks, weaknesses, and mitigations on which the assessment of risk level (i.e., likelihood x impact) is based. When the risks from the arguments are acceptable, the system is considered to have reached satisfactory security. As with this work, our assessment of risk of security requirements is based on information about potential attacks from CAPEC. Unlike this work, the assessment is based on non-compliance with the regulations. Other security assessment researches based on security catalogs are found also in different contexts, e.g., Piromsopa *et al.*[13] use web server vulnerability (or CVE) information from MITRE Corporation, issue HTTP requests to scan web servers to find vulnerabilities, and assess risk based on the the probability and impact of each vulnerability on web servers. Banklongsi and Senivongse[14] use information from CAPEC to define a security metric for web services based on the percentage of countermeasures provided against an attack type as well as severity, likelihood of exploit, and impact of the attack type.

4. Security Risk Assessment Method

The overview of the security risk assessment method is depicted in Fig. 1. We compile a standard set of security requirements from the IT Best Practices and match them with CAPEC attack patterns via the solutions and relevant security requirements of the patterns. The matching helps identify severity and likelihood of exploit of the attacks that might occur if the standard requirements are missing from the bank security requirements. We then calculate risk indices for the bank security requirements. The details are as follows.

4.1. Prepare Standard Security Requirements (SSRB)

First, we extract standard security requirements of banking (SSRB) from the IT Best Practices.[4, 5] There are 52 standard requirements under five domains (i.e., Core Banking Application, ATM Application Control, ATM Machine, Internet Banking Application Control, and Internet Banking Security). We give each requirement an ID for future reference and define a security category for each requirement. There are 12 categories[15]: Identification, Authentication, Authorization, Immunity, Integrity, Intrusion Detection, Non-repudiation, Privacy, Security Auditing, Survivability, Physical Protection, and System Maintenance Security.

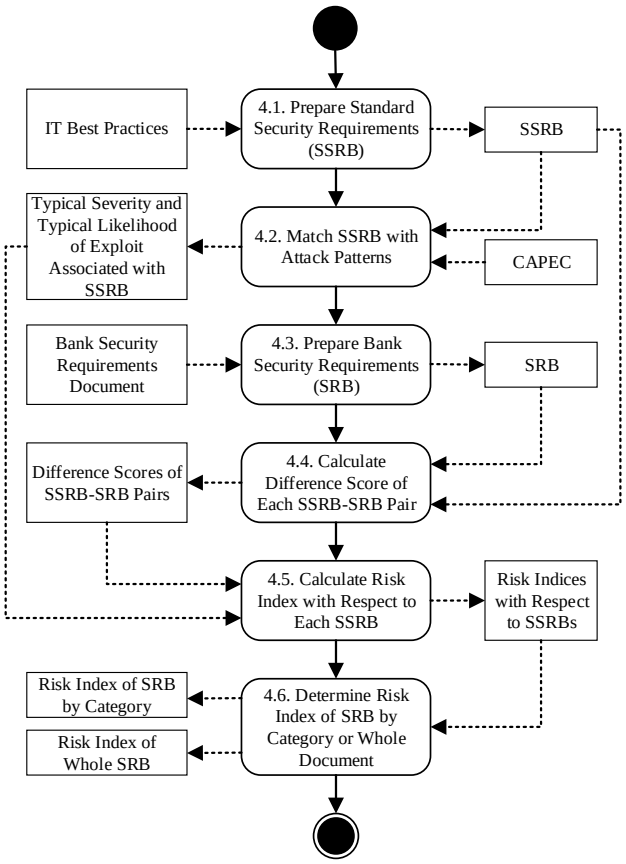

Fig. 1. Risk assessment method for the security requirements of a bank.

4.2. Match SSRB with Attack Patterns

In this step, we study the SSRBs and collect CAPEC attack patterns[6] that involve software and whose contents are related to the SSRBs. We filter out a number of attack patterns that address the implementation level of the software and keep 24 attack patterns whose details are at the requirement level. Table 1 shows an example. Since each SSRB is the security control that should be in place, we consider the Solutions and Mitigations and Relevant Security Requirements information in each attack pattern description to match them with the SSRB. For example:

SSRB_002: *The system shall enforce strong password (contain a mix of alphabetic and non-alphabetic characters).*

matches the Solutions and Mitigations of CAPEC_ID 16: *Create a strong password policy and ensure that your system enforces this policy.*

Table 1. Example of attack patterns.

CAPEC_ID	Attack Pattern Name	SEV	LOE
16	Dictionary-based Password Attack	High	Medium
49	Password Brute Forcing	High	Medium
50	Password Recovery Exploitation	High	Medium
60	Reusing Session IDs (aka Session Replay)	High	High
94	Man in the Middle Attack	Very High	Very High
169	Footprinting	Very Low	High
…	…	…	…

and the Solutions and Mitigations of CAPEC_ID 49: *Put together a strong password policy and make sure that all user created passwords comply with it.*

Therefore, the SSRB_002 and the CAPEC_ID 16 and 49 are identified as a match. We have the matching results reviewed by 12 BOT's security engineers and network engineers, with 2-10 years of experience. An example is in Table 2. We associate the Typical Severity (SEV) and Likelihood of Exploit (LOE) of the attack patterns with each SSRB. Typical Severity refers to the typical severity of impact on the software if this attack occurs. Typical Likelihood of Exploit means the likelihood of this attack typically succeeding considering the weakness attack surface, skills and resources required, available blocking solutions etc. We map the ordinal scale of Very Low, Low, Medium, High, and Very High of SEV and LOE to a numerical scale of 1-5 respectively for later calculation of the risk index. In the case that the SSRB is matched with more than one attack pattern, we use the principle of High Water Mark, i.e., the maximum level, to determine the SEV and LOE associated with the SSBR. For example, given Tables 1 and 2, the associated SEV of SSRB_004 is max(Very Low, High) = High = 4 and LOE is max(High, Medium) = High = 4.

4.3. Prepare Bank Security Requirements (SRB)

Given a security requirements document of a commercial bank, we prepare a list of bank security requirements (SRB), give each requirement an ID for future reference, and organize them into the same five domains as in the case of the SSRBs. An example is in Table 3.

Table 2. Example of SSRB and associated attack patterns.

SSRB_ID	SSRB Description	CAPEC_ID	SEV	LOE
SSRB_001	The system shall use strong encryption and security protocols to safeguard sensitive data during transmission over open, public networks.	94	5	5
SSRB_002	The system shall enforce strong password (contain a mix of alphabetic and non-alphabetic characters).	16, 49	4	3
SSRB_003	The system shall use two-factor authentication before password reset.	60	4	4
SSRB_004	The system shall re-authenticate when customer performs change of profile)e.g., address, telephone number, email(by hardware token.	169, 50	4	4
…	…	…	…	…

Table 3. Example of SRBs.

SRB_ID	SRB Description
SRB_001	The application shall use AES encryption and SSL protocol to safeguard during transmission over public networks.
SRB_002	The system will enforce a password to contain a mix of alphabetic and non-alphabetic characters and minimum password length is 8 characters.
SRB_003	The application shall use two-factor authentication (OTP) before password reset.
…	…

4.4. Calculate Difference Score of Each SSRB-SRB Pair

To determine how compliant the SRBs are with the SSRBs, we calculate the difference score of each SSRB and SRB requirement pair based on a text similarity measure. We preprocess the SSRB and SRB texts before similarity comparison as follows.

(i) *Segment words and remove stop words:* For SSRBs and SRBs, perform word segmentation on each requirement and remove stop words taken from a list of 619 words from http://countwordsfree.com /stopwords, plus 39 common domain words, e.g., system, application, customer.

(ii) *Change to lowercase letters:* Change all capitalized words to lowercase except for words with specific meanings, e.g., SSL, HTTP, SNA.

(iii) *Remove punctuation marks:* Remove punctuation marks, e.g., full stop (.), comma (,), brackets(()) etc.

(iv) *Remove suffixes:* Remove suffixes of words by truncating their suffixes using the Porter stemming algorithm.

(v) Determine difference between each requirement pair based on degree of similarity: Using the vector space model, we represent each SSRB and SRB requirement as a weighted term vector, where each element w_i of the vector is the weight of the term (or word) i that appears in that requirement. We follow the Cosine coefficient to determine similarity between each SSRB-SRB requirement pair by using the Term Frequency-Inverse Document Frequency weight (TF-IDF).[9] Since the similarity measure is bounded in [0, 1], it can be adapted to calculate the degree of difference $D_{q,r}$ between an SSRB requirement q from the IT Best Practices and an SRB requirement r of a bank by

$$D_{q,r} = 1 - \frac{\sum_{i=1}^{N} (w_{q,i} * w_{r,i})}{\sqrt{\sum_{i=1}^{N} w_{q,i}^2} * \sqrt{\sum_{i=1}^{N} w_{r,i}^2}} \quad (1)$$

where $w_{q,i}$ = weight of word i in SSRB requirement q,
$w_{r,i}$ = weight of word i in SRB requirement r,
N = number of distinct words in q and r, and
$D_{q,r}$ is in [0, 1].

Note that the following weight $w_{s,i}$ is used to determine the weight of word i in document s (i.e., each requirement q or r):

$$w_{s,i} = \begin{cases} tf_{s,i} * idf_i = (1 + \log_2 f_{s,i}) * \log_2 \frac{D}{d_i} & \text{if } f_{s,i} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where D = number of SRBs and
d_i = number of SRBs in which word i appears.

These $D_{q,r}$ values are calculated for every pair of SSRB and SRB requirements. Given the SSRBs in Table 2 and SRBs in Table 3, the difference scores are shown in Table 4.

4.5. Calculate Risk Index with Respect to Each SSRB

A risk index is a product of the probability of a risk (i.e., likelihood) and the severity of impact caused by the risk (i.e., consequence). In our context, there is the probability of risk of attacks associated with an SSRB requirement when it is not met by any SRBs of a bank. We represent this probability of risk by the minimum

Table 4. Example of difference scores of SSRB-SRB pairs.

SSRB_ID	SRB_ID	$D_{q,r}$
SSRB_001	SRB_001	0.3264
	SRB_002	1.0
	SRB_003	1.0
SSRB_002	SRB_001	1.0
	SRB_002	0.4116
	SRB_003	1.0
SSRB_003	SRB_001	1.0
	SRB_002	1.0
	SRB_003	0.0871
SSRB_004	SRB_001	1.0
	SRB_002	1.0
	SRB_003	1.0

$D_{q,ri}$ of each SSRB requirement q, i.e., the difference of the SRB requirement r_i that is the best match with the SSRB q. This probability is also weighted by the *LOE* of the attacks that could occur in the absence of that SSRB q from the banking information system. We use the *SEV* of those attacks as the severity impact caused by the risk. Thus, a risk index R_q with respect to any SSRB requirement q is calculated by

$$R_q = min(D_{q,r}) \times LOE_q \times SEV_q \quad (3)$$

where $min(D_{q,ri})$ = minimum difference score of an SSRB requirement q and any SRB requirement r_i,

LOE_q = Typical Likelihood of Exploit of attack patterns associated with q (e.g., *LOE* in Table 2),

SEV_q = Typical Severity of attack patterns associated with q (e.g., *SEV* in Table 2), and

R_q is in [0, 25].

For example, given Tables 4 and 2, the risk index with respect to each SSRB is in Table 5. We define a threshold (0.25 in this example) such that if the minimum difference score is greater than the threshold, we consider the SSRB *missing* from, i.e., not met by, the SRB requirements. Therefore, there is a degree of risk associated with the missing SSRB. On the other hand, if the minimum difference score falls below the threshold, we reset it to 0 and consider the SSRB *not missing* as an SRB that can meet that SSRB requirement is found.

4.6. Determine Risk Index of SRB by Category or Whole Document

In the previous section, an SRB that is the best match with an SSRB is identified, and a risk index is calculated with respect to how well the SRB can meet the SSRB. We then can determine the overall risk index for each of the 12 categories of the SRBs based on the risk indices associated with all SRBs under that category. Likewise, the overall risk index of the whole SRB document can be determined by the risk indices associated with all SRBs of a bank. Using the High Water Mark, we can represent the risk index R_c, where c is either a security category context or the whole document context, by the maximum risk index R_q associated with that context as in

$$R_c = max(R_q). \quad (4)$$

Given Table 5, the risk index by each category of the SRBs is shown in Table 6.

Table 5. Example of risk indices when threshold is 0.25.

SSRB_ID	SRB_ID	$min(D_{q,ri})$	SEV_q	LOE_q	R_q
SSRB_001	SRB_001	0.3264	5	5	8.16
SSRB_002	SRB_002	0.4116	4	3	4.9392
SSRB_003	SRB_003	0.0871 -> 0	4	4	0
SSRB_004	N/A	1.0	4	4	16

Table 6. Example of risk index by security category.

SSRB_ID	SRB_ID	Security Category	R_q	R_c
SSRB_001	SRB_001	Integrity	8.16	8.16
SSRB_002	SRB_002	Authentication	4.9392	4.9392
SSRB_003	SRB_003	Authentication	0	
SSRB_004	N/A	Identification	16	16

5. Evaluation

In this section, we report on the performance of compliance checking and on validity of the risk index.

5.1. Performance of Compliance Checking

We use the security requirements documents of nine commercial banks in Thailand to evaluate how well the proposed method can identify which SSRBs are missing from the SRBs of the banks. The method is applied to calculate the difference score of each SSRB-SRB requirement pair, find the SRB that is the best match, and determine if there are SSRBs that are not met by any SRBs with regard to a threshold. We use the results of bank requirements validation from the audits as the solution against which the performance of the method is evaluated. The solution tells which SSRBs are considered by the auditors as not met by, or missing from, the SRB documents of the banks.

Tables 7, 8, and 9 show the performance of the method in terms of F-measure of the two predicted classes of the SSRBs (i.e., Missing and Not Missing) and the overall accuracy respectively. The result is depicted graphically in Fig. 2. At the threshold of 0.4, all average performance reaches 80% and the best performance is when the threshold is 0.5. Starting at the threshold of 0.54, false negatives begin to show (i.e., the method predicts missing SSRBs as Not Missing) and the F-measure of the Missing class subsequently drops. We consider the false negatives as risky and not desirable, and so the threshold between 0.4-0.5 is recommended. Note that, at the threshold of 0.75, the F-measure of the Missing class for Doc#4, Doc#8 and Doc#9 is not applicable (N/A) because there is no SSRB with the difference score of at least 0.75 and predicted as Missing. So the precision, and hence the F-measure, of the Missing class cannot be calculated.

On taking a closer look at Doc#2 and Doc#5, we notice that the performance is particularly high even at the low threshold of 0.05-0.25, compared with the other seven documents. As the SSRBs usually mention technical security terms as the examples of the techniques that should be implemented, Doc#2 and Doc#5 which are written in more technical terms, match better with the SSRBs than the other seven documents which are written in more general terms and do not contain many technical terms. For example, an SSRB states that "*The system shall encrypt confidential information (e.g., user ID, password encryption key, database user id, database password) by strong encryption (e.g., AES 128 bits, AES 256 bits, RSA 2048).*" While Doc#1 is written as "*The application shall apply strong encryption to encrypt confidential information and store in a database or configuration file with appropriate access control.*", Doc#5 is written as "*The application shall use AES 256 bits encryption to encrypt system user id, password, database user id, and password before storing those hashes in a configuration file.*" The difference score of Doc#5 with more technical term is lower, and this goes along with the view of the auditors who are less likely to agree with a requirement as a general statement. The auditors usually expect the banks to be explicit about the techniques used whenever possible.

Table 7. F-Measure of Missing class.

Threshold	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5	Doc#6	Doc#7	Doc#8	Doc#9
0.05	52.17	80.00	11.11	50.00	83.87	57.14	66.67	57.14	40.00
0.15	70.59	85.71	14.29	57.14	89.66	61.54	66.67	61.54	45.45
0.25	85.71	92.31	20.00	66.67	92.86	66.67	76.92	50.00	55.56
0.30	92.31	92.31	22.22	77.78	96.55	66.67	76.92	54.55	71.43
0.40	100.00	92.31	40.00	80.00	96.30	66.67	76.92	60.00	100.00
0.50	90.91	100.00	100.00	92.31	96.30	72.73	90.91	60.00	75.00
0.55	80.00	95.65	100.00	92.31	96.30	60.00	90.91	44.44	75.00
0.65	66.67	58.82	100.00	80.00	78.26	57.14	80.00	57.14	33.33
0.75	50.00	40.00	100.00	N/A	63.16	40.00	66.67	N/A	N/A
0.85	28.57	15.38	N/A	N/A	26.67	N/A	28.57	N/A	N/A
0.95	28.57	N/A	N/A	N/A	14.29	N/A	N/A	N/A	N/A

Table 8. F-Measure of Not Missing class.

Threshold	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5	Doc#6	Doc#7	Doc#8	Doc#9
0.05	26.67	25.00	20.00	14.29	28.57	0.00	0.00	0.00	21.05
0.15	76.19	60.00	50.00	47.06	66.67	28.57	0.00	28.57	45.45
0.25	91.67	83.33	71.43	70.00	80.00	50.00	57.14	25.00	69.23
0.30	96.00	83.33	75.86	80.00	88.89	50.00	57.14	44.44	86.67
0.40	100.00	83.33	90.91	86.96	90.91	50.00	57.14	60.00	100.00
0.50	96.30	100.00	100.00	96.00	90.91	66.67	88.89	60.00	94.44
0.55	92.86	93.33	100.00	96.00	90.91	60.00	88.89	54.55	94.44
0.65	89.66	66.67	100.00	92.86	66.67	76.92	80.00	76.92	89.47
0.75	86.67	60.87	100.00	81.25	63.16	80.00	72.73	66.67	87.18
0.85	83.87	56.00	97.30	81.25	52.17	75.00	61.54	75.00	87.18
0.95	83.87	53.85	97.30	81.25	50.00	75.00	57.14	75.00	87.18

Table 9. Accuracy.

Threshold	Doc#1	Doc#2	Doc#3	Doc#4	Doc#5	Doc#6	Doc#7	Doc#8	Doc#9
0.05	42.11	68.42	15.79	36.84	73.68	40.00	50.00	28.00	31.82
0.15	73.68	78.95	36.84	52.63	84.21	50.00	50.00	40.00	45.45
0.25	89.47	89.47	57.89	68.42	89.47	60.00	70.00	56.00	63.64
0.30	94.74	89.47	63.16	73.68	89.47	60.00	70.00	72.00	81.82
0.40	100.00	89.47	84.21	84.21	94.74	60.00	70.00	88.00	100.00
0.50	94.74	100.00	100.00	94.74	94.74	70.00	90.00	80.00	90.91
0.55	89.47	94.74	100.00	94.74	94.74	60.00	90.00	80.00	90.91
0.65	84.21	63.16	100.00	89.47	73.68	70.00	80.00	72.00	81.82
0.75	78.95	52.63	100.00	68.42	63.16	70.00	70.00	68.00	77.27
0.85	73.68	42.11	94.74	68.42	42.11	60.00	50.00	68.00	77.27
0.95	73.68	36.84	94.74	68.42	36.84	60.00	40.00	68.00	77.27

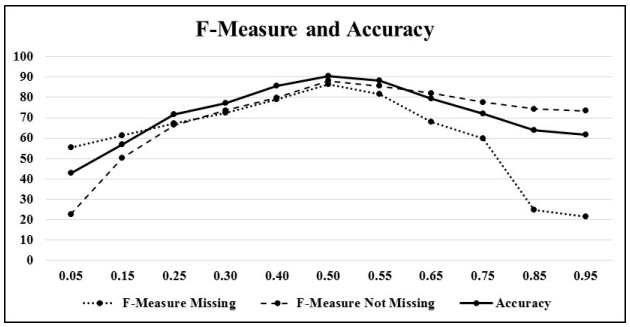

Fig. 2. Average F-measure and accuracy with regard to all SRB documents.

We further experiment by dividing the SRB documents into two groups, i.e., Doc#1, Doc#3, Doc#4, Doc#6, Doc#7, Doc#8 and Doc#9 that use less technical terms and Doc#2 and Doc#5 that use more technical terms. The performance with regard to the two groups is in Fig. 3 and Fig. 4. For the non-technical terms group, the best performance is still when the threshold is 0.5 as there are false negatives when the threshold is higher. For the technical terms group, the performance is best even at the low threshold of 0.25 and continues so until the threshold reaches 0.5 where the performance subsequently drops and false negatives appear. This experiment suggests the requirements engineers who use the proposed method to consider the writing style of the SRB document and may adjust the threshold accordingly.

5.2. Validity of Risk index

To validate the risk index, we use Spearman's rank order correlation to determine the correlation between the risk index of each SRB document as calculated by the method and the ordinal risk level determined by 12 security engineers and network engineers as in Table 10. We map the ordinal risk level of Very Low, Low, Medium, High, and Very High, given by the engineers, to a numerical scale of 1-5 respectively. For the risk index by the method whose value is in [0, 25], we map

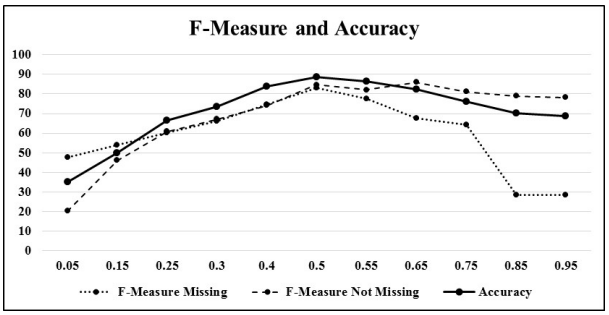

Fig. 3. Average F-measure and accuracy with regard to SRB documents that use less technical terms.

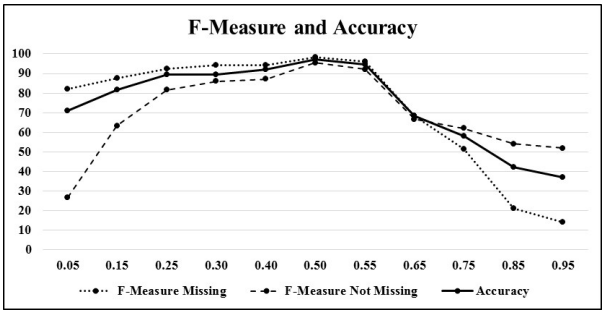

Fig. 4. Average F-measure and accuracy with regard to SRB documents that use more technical terms.

the range [0, 5] to 1, (5, 10] to 2, (10, 15] to 3, (15, 20] to 4, and (20, 25] to 5.

The hypotheses are

H_0: There is no monotonic correlation between the risk level by the engineers and the risk index by the method ($\rho_s = 0$).

H_1: There is a monotonic correlation between the risk level by the engineers and the risk index by the method ($\rho_s \neq 0$).

Table 10. Example of difference scores of SSRB-SRB pairs.

	Risk Level by Engineers	Mapped Risk Level by Engineers	Risk Index by Method	Mapped Risk Level by Method
Doc#1	High	4	16.00	4
Doc#2	Very High	5	20.52	5
Doc#3	High	4	13.17	3
Doc#4	Very High	5	15.55	4
Doc#5	Very High	5	20.55	5
Doc#6	Very High	5	16.23	4
Doc#7	Medium	3	15.87	4
Doc#8	Medium	3	14.79	3
Doc#9	Low	2	12.00	3

The calculated correlation coefficient $r_s = 0.74186$. Since r_s is not less than $r_{critical} = 0.7000$ at the significance level $\alpha = 0.05$ and n = 9, we reject H_0 and accept H_1. There is a monotonic correlation between the risk level by the engineers and the risk index by the method at $\alpha = 0.05$. The correlation is strong and positive.

To experiment further, we consider the total of 61 missing SSRBs from all nine SRB documents, together with their associated risk indices, to test a monotonic correlation with the risk levels given by the engineers. The distribution of the mapped risk levels of all 61 SSRBs is shown in Fig. 5. In this case, since $r_s = 0.63824$ is not less than $r_{critical} = 0.252$ at the significance level $\alpha = 0.05$ and n = 61, we again reject H_0 and accept H_1. For the case of missing SSRBs, there is also a monotonic correlation between the risk level by the engineers and the risk index by the method at $\alpha = 0.05$. Again, the correlation is strong and positive.

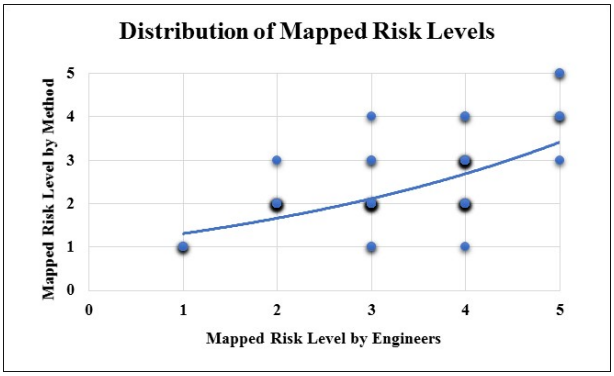

Fig. 5. Average F-measure and accuracy with regard to all SRB documents.

6. Conclusion

Given its satisfactory performance, the proposed automated risk assessment of security requirements of banking information systems is a useful approach to supporting requirements engineers or business analysts to check for compliance with security regulations. The method can help point out where the weaknesses are in the security requirements document and prioritize the improvement. Even though the context of this paper is the banking sector, we believe the method can be applied to other domains where checking requirements for compliance with regulations and attack-based security measurement are desirable.

To improve the method, the risk index model should be able to distinguish between terms in the standard security requirements which may indicate different prioritization categories, e.g., must, shall, should, could. In the case of BOT's IT Best Practices, all standard requirements use the term "shall" and therefore the current risk index model does not consider such requirement prioritization. However, if the method is applied to a different set of security regulations that might state different prioritization for different standard requirements, the risk index model should be enhanced since a missing requirement that is a must-have should be considered at a higher risk when compared to the missing of another requirement with lower importance.

To further improve the performance, we plan to experiment on using domain-specific stop words, semantics-related words, and spelling correction prior to the analysis. Human correction can also be allowed to adjust compliance checking and risk index results, e.g., based on specific environmental setting.

References

1. J. Lee, A View of Cloud Computing, *Int. J. Networked and Distributed Computing*, Vol. 1, No. 1 (2013), 2-8.
2. Y. Duan, X. Sun, A. Longo, Z. Lin, and S. Wan, Sorting Terms of "aaS" of Everything as a Service, *Int. J. Networked and Distributed Computing*, Vol. 4, No. 1 (2016), 32-44.
3. SecurityScorecard, *2016 Financial Industry Cybersecurity Report* (2016), https://cdn2.hubspot.net/hubfs/533449/SecurityScorecard_2016_Financial_Report.pdf, Accessed 20 May 2017.
4. Bank of Thailand, *IT Best Practices Phase I, Thailand* (2013).
5. Bank of Thailand, *IT Best Practices Phase II, Thailand* (2014).
6. The MITRE Corporation, *CAPEC–Common Attack Pattern Enumeration and Classification*, http://capec.mitre.org, Accessed 15 April 2017.
7. T. Li, E. Paja, J. Mylopoulos, J. Horkoff, and K. Beckers, Security attack analysis using attack patterns, in *Proc. 2016 IEEE 10th Int. Conf. Research Challenges in Information Science (RCIS)* (2016), pp.1-13
8. E. J. Steirna and N. C. Rowe, Applying information retrieval methods to software reuse A case study, *J. Inform. Process. and Manage.*, Vol. 39, No. 1 (2013) 67-74
9. R. Baeza-Yates and B. L. Ribeiro-Neto, *Modern Information Retrieval*, 2nd ed (ACM Press, New York, 2011).
10. M. Ilyas and J. Kung, A similarity measurement framework for requirements engineering, in *Proc. 2009 4th Int. Multi-Conf. Computing in the Global Inform. Technology* (Cannes, La Bocca, 2009), pp. 31-34.
11. J. N. O. Dag, B. Regnell, P. Carlshamre, M. Andersson, and J. Karlsson, A feasibility study of automated natural language requirements analysis in market-driven development, *J. Requirements Eng.* **7**(1) (2002), 20-33.
12. Y. Yu, V. N. L. Franqueira, T. T. Tun, R. J. Wieringa, and B. Nuseibeh, Automated analysis of security requirements through risk-based argumentation, *J. Syst. and Software*, Vol. 106 (2015) 102-116.
13. K. Piromsopa, T. Rojkangsadan, and N. Prompoon, A Risk assessment of web server impact classification by loss type, in *Proc. Networks and Commun. Syst. (NCS)* (2005), pp. 173-178.
14. T. Banklongsi and T. Senivongse, A security measurement model for web services based on provision of attack countermeasure, in *Proc. 15th Int. Annu. Symp. Computational Sci. and Eng. (ANSCSE15)* (2011), pp. 593-598.
15. D. G. Firesmith, Engineering security requirements, *J. Object Technology*, Vol. 2754 (2003), 53-68.

.