The Journal of Systems & Software 169 (2020) 110695

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Understanding and recommending security requirements from problem domain ontology: A cognitive three-layered approach

Bong-Jae Kim [a], Seok-Won Lee [b,*]

[a] Department of Network Centric Warfare, Ajou University, Suwon City, 443-749, Republic of Korea
[b] Department of Software and Computer Engineering, Ajou University, Suwon City, 443-749, Republic of Korea

ARTICLE INFO

Article history:
Received 12 September 2019
Received in revised form 3 June 2020
Accepted 8 June 2020
Available online 18 June 2020

Keywords:
Security
Requirements engineering
Ontology

ABSTRACT

Socio-technical systems (STS) are inherently complex due to the heterogeneity of its intertwined components. Therefore, ensuring STS security continues to pose significant challenges. Persistent security issues in STS are extremely critical to address as threats to security can affect entire enterprises, resulting in significant recovery costs. A profound understanding of the problems across multiple dimensions of STS is the key in addressing such security issues. However, we lack a systematic acquisition of the scattered knowledge related to design, development, and execution of STS. In this work, we methodologically analyze security issues from a requirements engineering perspective. We propose a cognitive three-layered framework integrating various modeling methodologies and knowledge sources related to security. This framework helps in understanding essential components of security and making recommendations of security requirements regarding threat analyses and risk assessments using Problem Domain Ontology (PDO) knowledge base. We also provide tool support for our framework. With the goal-oriented security reference model, we demonstrate how security requirements are recommended based on PDO, with the help of the tool. The organized acquisition of knowledge from SME groups and the domain working group provides rich context of security requirements, and also enhances the re-usability of the knowledge set.

1. Introduction

1.1. Motivation

Socio-technical Systems (STS) involve relationships between hardware, software, and humans (Baxter and Sommerville, 2011). System requirements for STS are inherently complex due to involvement of diverse stakeholders. The complexity is mainly two-fold: qualitative and quantitative. Qualitative complexity comes from interactions among businesses, humans, technologies, and environments; quantitative complexity comes from the increasing sizes of networks, systems, and businesses. The security issues derived from these complexities are still widely considered problematic, and potential and real security intrusions affect the entire industry. As seen in the recent attack cases, for example, by APT39 (FireEye, Inc, 2019), an Iranian cyber espionage group, and APT41 (FireEye, Inc, 2020), a prolific cyber threat group responsible for Chinese state-sponsored espionage and financially motivated activities, Advanced Persistent Threats (APT) affect the entire STS. APT39 attacks mainly telecommunications and travel industries. The targets store large amounts of personal and customer information. Those data cannot only provide access to critical infrastructures but also enable access to a wide range of extended targets. APT41, in case of financially motivated activity, compromises the software supply chain to enable them to inject malicious code into legitimate files. The injected code can be used to compromise additional organizations. These attacks are carried out along the APT lifecycle, using advanced and complex techniques, and impact on a wide range.

According to a survey by the Phonemon Institute in 2011 (Ponemon, 2011), the average recovery cost from APT was estimated as approximately $5.5 million per case. In order to solve these security issues, researchers have made significant efforts in enhancing the security quality attributes of STS, and in application of these enhancements. Despite these efforts, however, security incidents have occurred consistently, causing financial, business, and social problems (FireEye, Inc, 2019, 2020).

As it is believed that the key to identifying more effective solutions involves examination of the problem space, we have investigated the STS security issues from a Requirements Engineering (RE) perspective. Security requirements should be considered and applied to STS in early phases of its design. Despite proposals of various methods for modeling and analysis of security requirements from requirements engineering researchers (Fabian et al., 2010), these methods fail to reflect the diverse perspectives of

* Corresponding author.
E-mail address: leesw@ajou.ac.kr (S.-W. Lee).

https://doi.org/10.1016/j.jss.2020.110695
0164-1212/

understanding the security environments of STS: the complexity of consolidating and organizing the knowledge of development and execution of STS has not been addressed well by recent research efforts. Such knowledge is scattered or fragmented into a variety of knowledge sources, and highly interdisciplinary, involving work in business, law, and system services (Pennock and Wade, 2015; Collopy, 2012). In addition, threats targeting STS are in continuous evolution. These issues exponentially increase the complexity, rendering systematic organization of security-related knowledge difficult.

Consideration of stakeholders' areas of expertise in generation of the security requirements is crucial, especially considering security-quality attributes and their rippling negative effects on other quality attributes (Elahi, 2009). As seen in the above examples by APT groups, APT attacks impact on a wide range of targets. In the case of the attack by APT39, their targets are expanding to various areas such as telecommunications, travel industry, IT firms, high-tech industry, transportation, and government entities (FireEye, Inc, 2019). The environments in STS include a large number of stakeholders in various areas of expertise. This fact shows the increasing complexity of the interdependencies among the quality attributes of security requirements and diverse stakeholders being involved in various APT attacks. This means that the security requirements, against APT attacks such as activities by APT39 or APT41, have to reflect the requirements of these stakeholders in various areas of expertise. Therefore, it is necessary to generate security requirements for STS systemically with a comprehensive view of diverse stakeholders. However, it is hard to satisfy various stakeholders when security requirements are merely generated from a technical analysis or modeling approach. Without proper consideration of stakeholder expertise, critical defects in requirements can occur, and could limit the generation of explicit security requirements for STS development and operation.

A survey by Elahi et al. (2011) on the usage of knowledge shows, interestingly, requirements engineers do not use knowledge from a wide range of sources during the requirements engineering process. In generation of the security requirements for STS, organization and utilization of knowledge from various sources is crucial, and some of the knowledge must be reusable. The survey demonstrates that 42% of responses rarely use standards or Common Criteria (CC) (Common Criteria, 2012), and the rest of respondents only use CC and domain standards in generation of security requirements. As a result, the generated security requirements are specified to the experience level of the Subject Matter Expert (SME), which means maintenance and re-application of the requirements pose challenges.

Therefore, we focus on solving the security requirements related issues of STS in a knowledge-intensive way. The goal of this research is to provide guidance to organize the scattered STS related knowledge and models, and to perform the requirements engineering activities efficiently by understanding and recommending the security requirements with the specification. To achieve this goal, we derive the specific propositions mainly focusing on the following three aspects of the research problem. Helping users to:

(1) Organize various sets of related documents and relevant modeling approaches.

(2) Understand the complex security environment of the targeted STS.

(3) Recommend security requirements easily.

1.2. Approach and proposed idea

We have developed a framework for understanding and recommending security requirements with threat analyses and risk assessments using a Problem Domain Ontology (PDO) knowledge base. Based on our aims, we propose the Security requirements Physical, Information-modeling and Cognitive layered framework (SPIC framework) for understanding and recommending security requirements using a cognitive layered approach (Smith, 2006) and extended PDO for security requirements (Lee et al., 2006).

PDO provides agreement, common understanding, and explicit information sharing among related stakeholders using various knowledge sources and enables reuse of knowledge. The SPIC framework consists of a physical layer, an information-modeling layer, and a cognitive layer, and contains various meta-models for reasoning the security issues using the PDO. This framework helps to understand security environments and recommendations for security requirements for related STS. Furthermore, we propose a knowledge framework coupled with knowledge from various sources using specific security goals in order to analyze threats and assess risks. The SPIC framework enhances knowledge reusability by dividing the knowledge into General Purpose Knowledge Base (GPK) and Domain Specific Knowledge Base (DSK).

The remainder of this paper is organized as follows: Section 2 discusses related work and background information. Section 3 introduces the proposed framework with a running example of a real threat scenario. In Section 4 we explain the support tools. Finally, we evaluate the proposed framework in Section 5 and conclude with potential future directions of our work in Section 6.

2. Related work and background

2.1. Security requirements engineering modeling methodology

A number of security requirements engineering researchers have proposed various modeling methodologies regarding security requirements. We compared these modeling methodologies been referred to in categorizations by Fabian et al. (2010) and Elahi (2009), Elahi and Yu (2007). These modeling methodologies were analyzed in four components: the modeling elements, knowledge sources, model reusability, and specification methods. Modeling elements are the elements for modeling in order to understand security environment. These can include involving an AS (Asset), ST (Stakeholder), EN (Environment), TH (Threat), VU (Vulnerability), RI (Risk), SG (Security Goal: to explain 'why'), SR (Security Requirement: to explain 'how') and CM (Countermeasure). Knowledge sources are the required knowledge type for modeling. These are represented in two categories: SME K (which depends on SME Knowledge) and Standard K (depends on Standardized knowledge). Model reusability is the scope for reusing models or knowledge. Methodologies can be either DS (Domain Specific) and GP (General Purpose). If the models or knowledge in a modeling methodology can be reused in particular domain, the"Model Reusability" of this methodology is"DS". CORAS, for example, is a security risk analysis model (Den Braber et al., 2006) proposed by SINTEF, a European independent research organization, in 2006. It can be customized and used in particular domain and simultaneously modeled and reused. The final base for analyzing methodology is the specification method. This refers to a proposed specification guidance in the research. The categories for specification methods are methodologies are those that: T (use a template), C (use diagram elements), and N (use Natural Language Specifications). Table 1 shows a comparison of these modeling methodologies based on these analysis units. For example, MSRA (Multilateral Security Requirements Analysis) is the security requirement methodology in a multilateral approach. The objective of MSRA is to apply the principle of multilateral security in the requirements engineering phase of systems development (Gürses et al., 2006). Multilateral security

Table 1
Comparison of the modeling methodologies for security requirements.

Modeling	Modeling Elements	Knowledge Sources	Model Reusability	Specification Method
MSRA Gürses et al. (2006)	AS, ST, SG, EN	SME K	DS	T
SQUARE Mead and Stehney (2005)	ST, TH, SG, RI	SME K	DS	N
Misuse case Sindre and Opdahl (2001)	SG, TH, VU, RI, ST, AS	SME K	DS	T
Secure UML Lodderstedt et al. (2002)	ST, EN	SME K	DS	C
UMLsec Jürjens (2005)	SG, TH, VU, RI, ST, EN	SME K	DS	T
KAOS anti-model Van Lamsweerde (2004), Mellado et al. (2006)	SG, TH, VU, ST, EN, AS	SME K	DS	C
Secure Tropos Mouratidis and Giorgini (2007)	SG, TH, VU, RI, ST, SR	SME K	DS	C,T
GBRAM Antón and Earp (2001)	SG, TH, VU, RI, ST, AS	SME K	DS	C
STS FW Paja et al. (2015)	SG, ST, EN	SME K	DS	C
Three layered FW Li and Horkoff (2014)	SG, TH, VU, AS	SME K	DS	C
CORAS Den Braber et al. (2006)	TH, VU, RI, EN, AS, CM	SME K	DS	C
Tropos goal-risk FW Asnar et al. (2007)	SG, TH, RI, ST	SME K	DS	C
CC Common Criteria (2012)	SR, TH, VU, RI, SG, ST, EN, AS, CM	Standard K	GP	T
SREP Mellado et al. (2006)	SG, TH, VU, RI, EN, AS	Standard K	GP/DS	T

focusing on stakeholder views, and the circumstances of security requirements are important in MSRA (Fabian et al., 2010). In this methodology, security goals considering assets, all stakeholders, and environment are identified and described. Therefore, AS, ST, SG, and EN can be Modeling Elements. As MSRA is processed by SME (Subject Matter Expert) and can be reused in a domain, Knowledge Sources are categorized as "SME K", and "Model Reusability is "DS". The Specification Method for MSRA is "T", as MSRA a provides template.

This result shows the elements in each methodology are modeled differently, which means that it is difficult to understand the security requirements using only one methodology. Therefore, integration or extension of several models is required to fully understand the security situation in a domain. Even though CC or the Security Requirements Engineering Process (SREP) can generate the standard level of security requirements, the quality of models produced by other methodologies is determined by the level of knowledge and experience of the SMEs. In addition, even though the majority of the methodologies can reuse the produced models and knowledge within the domain or conditional circumstances, it is hard to reuse them in different circumstances. Moreover, the specification method is not explicitly provided by each methodology. Based on this result, we define the framework requirements of our proposed model as follows. The model must:

- Integrate various modeling elements and define the relationship
- Organize knowledge so that there is less dependence on subjective knowledge
- Enhance reusability of knowledge and models
- Provide the specification method

2.2. Research background

2.2.1. Cognitive domain research

The Effects-based Operation (EBO) proposed by Smith (2006) used a layered approach to represent the information relationship using physical, information, and cognitive domains. This approach was applied to the Observe–Orient–Decide–Act (OODA Loop) (Ford, 2010), which illustrates the decision-making process during military operation. The author proposed a methodology for modeling and analyzing the events in the physical layer into models and information in the information layer by systems or experts. As a result, the commander can recognize a situational context using the relationship between sets of information. We apply this cognitive layered approach to the SPIC framework.

2.2.2. Knowledge classification

Knowledge is the information and understanding of a subject people have Naver Corporation (2016). Understanding comes from cognitive activities, including cognition, communication, inference, etc. Michael Polanyi classified knowledge into two types: explicit knowledge and tacit knowledge (Grant, 2007). These types of knowledge are implemented in the proposed framework as general-purpose knowledge and domain-specific knowledge. General-purpose knowledge is explicit, reusable, and knowledge that is agreed upon across the security domain, such as methodology classification, categorization, principles, etc. Domain-specific knowledge includes tacit knowledge, embedded in corporate process and routines, and detailed knowledge applied in a specific area. Using the available knowledge, we build a General-Purpose Knowledge Base (GPK) and Domain-Specific Knowledge Base (DSK) with the framework.

2.2.3. Problem domain ontology for security requirements

Lee et al. (2006) proposed a methodology to build problem domain ontology using Onto-ActRE (Lee and Gandhi, 2005) and used it to generate security requirements. Their research applied ontology to deal with scattered knowledge by considering the relationship between concepts using a Defense Information Technology Security Certification and Accreditation Process (DITSCAP). Their methodology is applied to the SPIC framework to build the knowledge base in a systematic way. Kim and Lee (2015) proposed a methodology to build ontology using a goal model for analyzing the threat using a cognitive layered model.

Based on these works, we define the concepts for composing security requirements with reference to the CC (Common Criteria) general model. CC is the international standard for computer security evaluation. It provides a common set of requirements for the security functionality of IT products. CC describes security using a set of security concepts and terminology (Common Criteria, 2012). The concepts are not only quite general, but also easy to use. As shown in Fig. 1, the security requirements concept relationship model consists of assets, threats, vulnerabilities, security goal, malicious goal, risk, countermeasures, and security requirements. In addition, we define the security requirements specification elements based on the security requirements concept relationship model with Quality Attribute (QA) scenario components (Bass et al., 2012). A QA scenario is composed of source, stimulus, artifacts, environments, response and response measure. Security requirements concept relationship model components can be corresponded to these QA scenario components, as shown in Table 2.

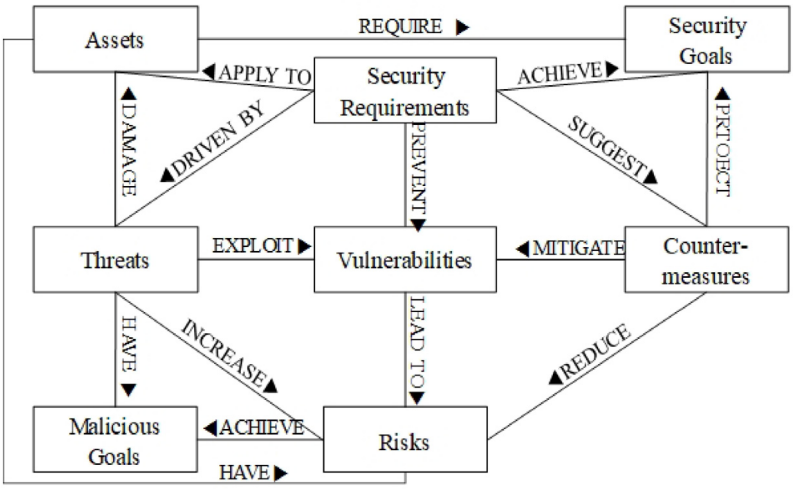


Fig. 1. Security requirements concept relationship model.

Table 2
Relationship between QA scenario components and Security Requirements Components.

QA Scenario Component	Security Requirements Components
Source	Threat, Attack Vector
Stimulus	Vulnerability
Artifacts	Asset
Environments	Asset Operating Environments, Stakeholders
Response	Countermeasure
Response Measure	Security Goal, Malicious Goal, Risk

2.2.4. Risk assessments

In this paper, we specify the recommended security requirements from risk assessment using the created knowledge base, Problem Domain Ontology (PDO). There are a number of methodologies for risk assessment. Blank and Gallagher (2012) of the National Institute of Standards and Technology (NIST) published research providing guidance for risk assessment, which mentioned processes to identify threats, systems, and vulnerabilities, and assess the risk based on these identified factors. Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) (Alberts et al., 2003; Caralli et al., 2007) was proposed by the Software Engineering Institute in Carnegie Mellon University and utilized for risk assessment using practical knowledge in an enterprise. ISO/IEC27005 is the international standard for information system risk management. It provides guidelines for information security risk management in an organization (ISO/IEC, 2018). Compared to other risk methodologies, it does not provide any specific method for information security risk management. The organization can define their approach to risk management, depending on the scope of an information security management system (ISMS), context of risk management, industry sector, and so on. As shown in Tables 1 and 8, a number of methodologies for security requirements include risk elements for modeling in order to understand security environments. These methodologies provide the insight to understand the relationship between concepts and processes for the risk assessment.

3. Introduction of the security requirements PIC framework

3.1. Overview of the framework

This chapter introduces the SPIC framework. This framework utilizes a goal-oriented modeling approach for threat analysis and risk assessments in order to understand and recommend the security requirements using PDO which scattered knowledge is well-organized.

3.1.1. Conceptual model

We introduce the SPIC framework, which consists of three layers: physical layer, information-modeling layer, and cognitive layer, as shown in Fig. 2. The physical layer contains real events, incidents, data, and knowledge sources. The information-modeling layer contains models and categorizations obtained by modeling, analyzing, categorizing, and classifying knowledge sources in the physical layer. The cognitive layer provides a comprehensive understanding and awareness based on the relationship among concepts, knowledge, and models. The main idea of this framework is that the security requirements can be understood and recommended with related specified concepts using PDO in the cognitive layer.

As shown in Fig. 2, phase 1 builds the explicit knowledge base GPK, which is reusable in the overall domain. Phase 2 builds the DSK, which integrates knowledge about architectures, threats and risks in the specific domain, and puts the two knowledge bases together into PDO using the relationship between concepts. Phase 3 provides the security requirements using PDO for the specific domain.

3.1.2. Usage of framework

This framework recommends security requirements with regard to general-purpose knowledge and domain-specific knowledge using the relationships between concepts. As described in the analysis in Section 2, security requirements engineering modeling methodologies depend on the knowledge level of SMEs to generate security requirements. However, the SPIC framework uses the GPK built by agreements among SMEs, which reduces variation in the level of knowledge in practice. It also utilizes DSK, built from domain-specific architecture, threats, and risks, to recommend security requirements. Even though working groups in an enterprise lack knowledge of security requirements, they elicit security requirements based on these two knowledge bases. Fig. 2 represents the actors of each phase and artifacts. The SME group performs phase 1 to build the GPK, which defines the relationship between categorizations and models from various knowledge sources. The domain-working group in each enterprise performs phase 2 to build DSK. After that, the working group builds the PDO based on GPK and DSK. Finally, they specify the recommended security requirements from risk assessments using

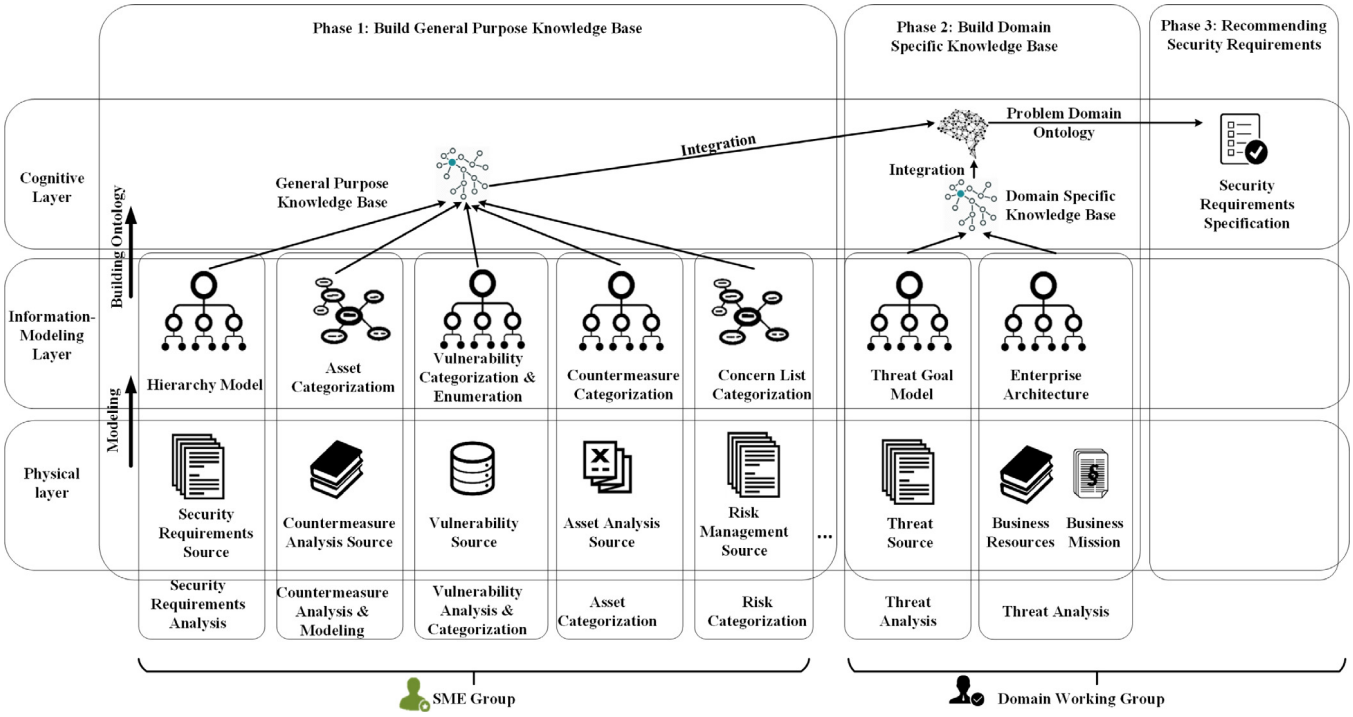


Fig. 2. Conceptual model for understanding and recommending the security requirements from the PDO.

the created PDO. Phases 2 and 3 are operated independently of phase 1 when the enterprise does not support a large SME group. Even though GPK is reusable across the entire security area, DSK from phases 2 and 3 is applied within the specific domain of the enterprise.

3.1.3. Output from framework

PDO, one of the outputs of SPIC framework, is a knowledge base that integrates GPK and DSK. GPK contains the agreed, hierarchical, and categorized knowledge about concepts, which enhances the reusability across the entire security area. DSK contains the domain specific and instantiated knowledge applied to domains or enterprises. Fig. 3 shows the frame model for the implementation of PDO. This model consists of ontology level and database level and is based on the risk and security requirements concept model shown in Fig. 1. The elements in Fig. 3 are defined as concepts and their attributes with the associated specification and categorization. In particular, the concepts and attributes that need to be in agreeance within the SME group are included in the GPK. Other concepts and attributes, different for each domain or enterprise, are included in the DSK. The relationship model is implemented at the ontology level. The contents of each concept and attribute are established in the database level. For example, Assets in the ontology level are mapped to TBL_PO_AS in the database level as GPK related attributes and simultaneously, are mapped to TBL_DO_DOMAINS, TBL_DO_DOMAINSPLATFORM, TBL_DO_RELATEDSTAKEHOLDER in the database level as DSK related attributes.

The SPIC framework provides security SPIC diagram and template to specify security requirements and is based on the elements of QA scenario components as shown in Table 2 (Bass et al., 2012). Fig. 4 shows a diagram of the SPIC framework and template. After processing risk assessment via the SPIC framework, the results can be represented with a diagram and template. A SPIC diagram includes elements specifying security requirements such as the security environment, assets, security goals (Confidentiality, Integrity, Availability), vulnerabilities, threats, attack vectors, countermeasures (Monitoring, Prevention, Recovery) and stakeholders. The diagram helps understand results of risk assessment and elicitation of security requirements. During the risk assessment process, the detailed data generated is represented in a SPIC template. This template include all elements involved in assessing risk, such as assets, security goals, threats, vulnerabilities, malicious goals (Exposure, Modification, Destroy), current countermeasures, risk factors, and recommended security requirements. Security requirements are specified and recommended through the process of risk assessment with these diagram and template in SPIC framework. These diagram and template can be customized by domains or enterprises.

3.2. SPIC framework process

The process of the SPIC framework, shown in Fig. 5, consists of three phases: build GPK, build DSK, and recommend security requirements. Fig. 5 and Table 3 show the introduction of the SPIC framework process. A large company is capable of proceeding with the SPIC framework from phase 1 to 3. However, smaller companies are only capable of performing phases 2 and 3 of the framework, with a GPK built by a general SME group to recommend the security requirements, due to smaller revenues. Based on the knowledge from the domain or enterprise, the SME group develops and maintains the GPK. DSK and PDO are constantly developed and maintained based on the SPIC framework within the domain and enterprise.

3.2.1. Brief overview of the case study

For better understandability, we explain the framework process step-by-step in the next sub-section with the help of a case study.

Threat Scenario: Butterfly (Hacktool.Eventlog)

In this case study, a real threat scenario involving Hacktool.Eventlog is posed. This tool has been used by Butterfly since 2012 for intrusion against technology or pharmaceutical companies. Butterfly focused on espionage against industrial and technological companies from the late 2012 to early 2013 and currently conducts target finding, intrusions, and attacks using

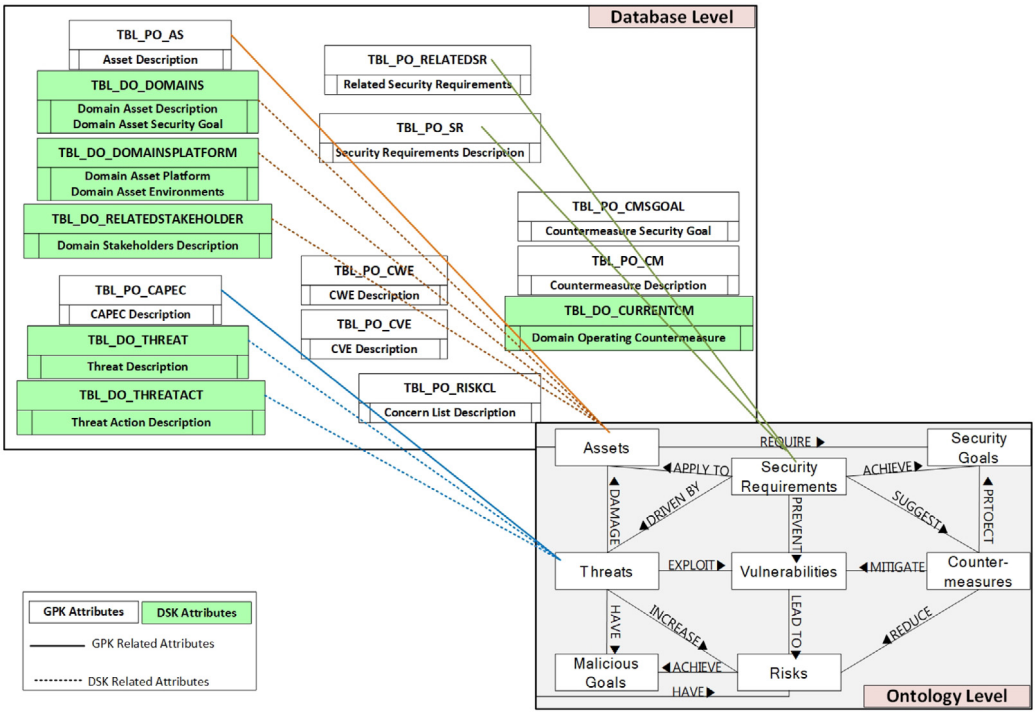
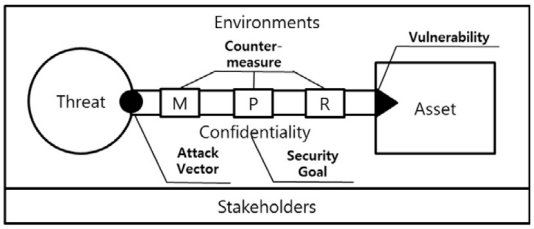


Fig. 3. Problem domain ontology frame model for implementation-ontology level and database level.



Asset	Asset Security Goal			Threat	Vulnerability	Malicious Goal			Current Countermeasures									Risk Factor			Recommendations
									C			I			A						
	C	I	A			E	M	D	M	P	R	M	P	R	M	P	R	E	M	D	

Fig. 4. SPIC Specification Diagram and Template.

various tactics, techniques, and zero-day vulnerabilities in order to achieve malicious goals. The difference from that of previous espionage cases is that Butterfly focuses more on financial purposes to conduct an attack. The main targets for intrusion are e-mail servers or enterprise content management servers. The main tactic for conducting an attack is to concentrate on a specific target that satisfies the environment, platform, and configuration after accessing and retrieving information by finding one or more zero-day vulnerabilities. In particular, Hacktool.Eventlog parses, dumps, and deletes log data including self-destruction. The case study focuses on Hacktool.Eventlog by illustrating the process of building GPK, building DSK, and recommending security requirements. We use Security Response Report by Symantec (Symantec Security Response, 2015) for analyzing the threat, and assume that this is a zero-day vulnerability even though it has already been solved.

Environment Introduction

We developed a virtual environment for applying the SPIC framework. Since Butterfly aims to attack IT companies for financial reasons, we set a virtual environment with a mobile software development company. The name of the company is BK, which develops mobile software and has several teams: Development, General Affairs, Accounts, and System Management. Fig. 6 shows the virtual network of this company. The company has its own DB, Web and E-mail server with firewalls in the company network. The current company security measures are shown in Table 4 with a partial summary. Security measures for network traffic assets can be considered with countermeasure type such as monitoring, preventing, and recovery . In the case of monitoring, as shown in Table 4, there are "IDS/IPS" and "Router/Access Control" security measure for security types — Confidentiality (C) and Availability (A), and security measure "Firewall" for security type Availability (A). The security service column shows related security services for each security measure.

3.2.2. Phase 1: Building general purpose knowledge base

This phase builds the GPK based on the agreeance in the SME group by following the customized methodology in Lee et al. (2006) and Lee and Gandhi (2005). The GPK is produced from universally applied security knowledge sources,like documents and categorization using the relationship between concepts. This phase consists of three sub-steps to build the GPK (Table 3).

Preparation

The preparation step identifies concepts, attributes, and related knowledge sources for security requirements. The concepts

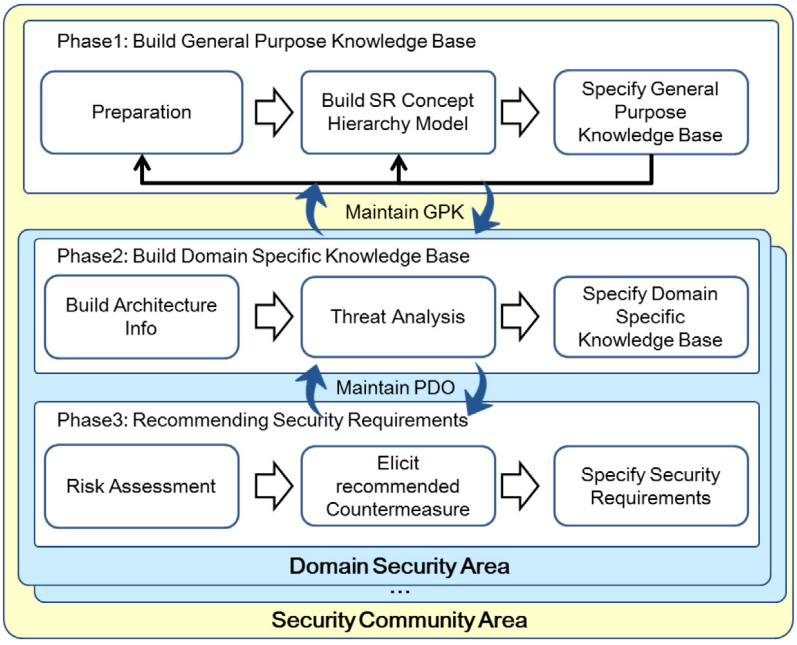


Fig. 5. SPIC framework process model and practice scope.

Table 3
SPIC framework process model description.

Phase	Step	Activity	Input	Output
Build GPK	Preparation	Define security requirements concepts, attributes, and knowledge sources	Knowledge sources	Identified security requirements concepts, attributes and knowledge sources.
	Build SR Concept Hierarchy Model	–Create classification and categorization for concepts –Provide ID number to concepts and attributes –Create Security requirements concepts relations	Output of preparation	Requirements concept hierarchy and relation
	Specify GPK	–Specifying concepts and attributes –Build General Purpose Knowledge Base	Output of Preparation and Build SR Concept Hierarchy Model	General Purpose Knowledge Base
Build DSK	Build Architecture Info	Create Architecture Info related to security concern	Architecture knowledge sources	Architecture information.
	Threat Analysis	–Analyze threat related to the domain area	Threat knowledge sources	attack goal based model
	Specify DSK	–Build Domain Specific Knowledge Base	Architecture Information and Attack Goal Based Model	Domain Specific Knowledge Base, and PDO
Recommending SR	Risk Assessments	Assess risks based on PDO	PDO	Identified risk factor.
	Elicit Recommending Countermeasure	Elicit recommended countermeasure based on identified risk factor	Risk factor and related concepts	Recommended countermeasure
	Specify SR	Specify recommended security requirements	Related concepts	SPICS diagram and template

and attributes are identified using related knowledge sources. The concepts and relationships between concepts are shown in Fig. 1. Attributes are divided into two types, shared attributes and dependent attributes. The shared attribute is shared with multiple concepts, such as malicious goal and attack vector. The dependent attribute is applied only to a single concept, such as the stakeholder and description (of each). Table 5 shows an example list of the concept and attribute for SPIC framework. for example, in case of the concept, “Assets", its attributes are description, related stakeholders, security goal, and concern list. These concepts and attributes are adjustable according to the domain or enterprise.

It is necessary to collect related materials during preparation, including documents from the agency or the security company. We use ISMS Certification and Accreditation Process Guidance (Korea Internet and Security Agency, 2013), ISMS Risk

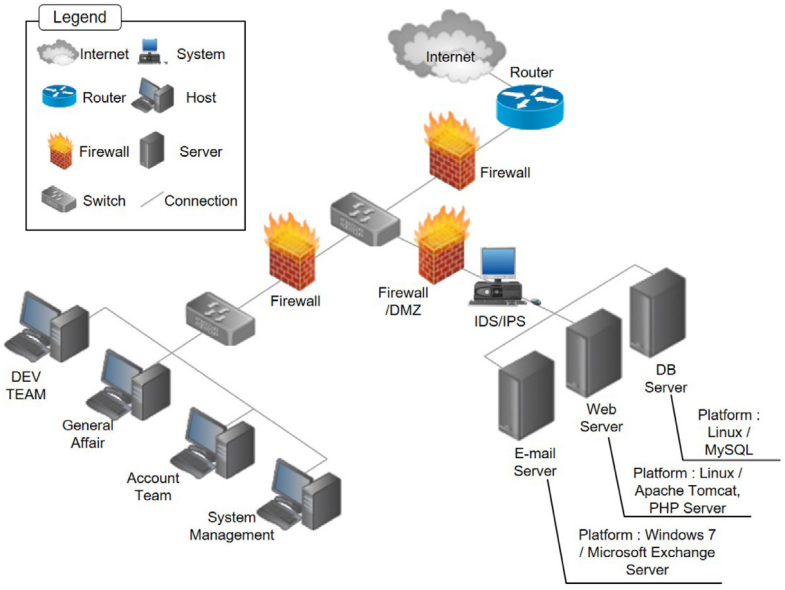


Fig. 6. Virtual network topology for the case study.

Table 4
Partial summary for operational security countermeasures in the BK Company.

Asset		Counter- measure Type	S. A.	Security Measure	Security Service
Network	Traffic	Monitoring	C	IDS/IPS	• Access Control. • Traffic Control.
				Router/Access Control	• Access Control
			A	Firewall	• Detect the DDoS Attack.
		Preventing	C	IDS/IPS	• Access Control and Drop the packet.
				TLS/SSL Certification	• Provide Secure Communication
			A	DMZ Network	• Provide separated zone for server and inner network.
		Recovery	A	Server Backup Port Policy	• Provide the backup port in case of unavailable port.
Data	Log Data	Monitoring	C	Server/Access Control	• Monitor user who access to the log file.
			A	Log Review	• Review the Log data Periodically.
		Preventing	C	Server/Access Control	• Prevent unauthorized user from accessing the log data.
				Application/Security Feature	• Prevent users to exposure security error based on error log data
			A	Log Review	• Review the Log data Periodically.
		Recovery	A	Server Backup Port Policy	• Back up the Log data into external media.
	DB Data	Monitoring	C	Server/Access Control	• Monitor user who access to the Database.
		Preventing	C	DB Encryption	• Encrypt Data not to be exposed information by unauthorized user.
			I	DB Encryption	• Prevent modification of data without decryption key.
		Recovery	A	DB Backup	• Back up the Database data.

Management Guidance (Korea Internet and Security Agency, 2004) by KISA, Common Weakness Enumeration (CWE) (MITRE corporation, 2020), Common Attack Pattern Enumeration and Classification (CAPEC) (MITRE Corporation, 2020a), Common Vulnerability and Exposure (CVE) (MITRE Corporation, 2020b), Common Platform Enumeration (CPE) (MITRE Corporation, 2014) by MITRE Corporation, and law information from Korea (Ministry of Government Legislation, 2016). With these sources of information,

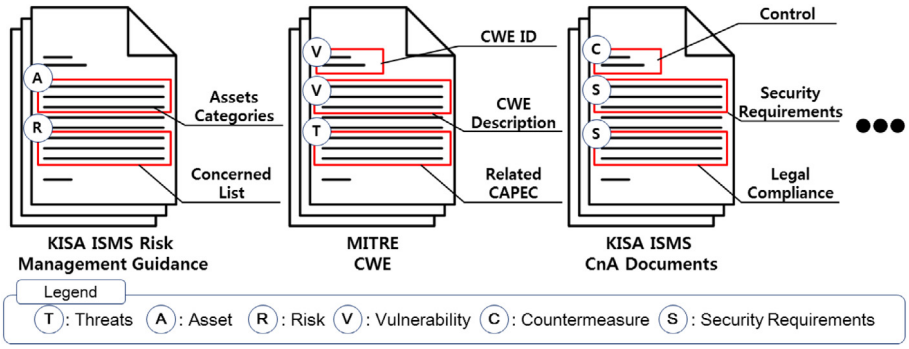


Fig. 7. Identify security requirements concepts from knowledge sources.

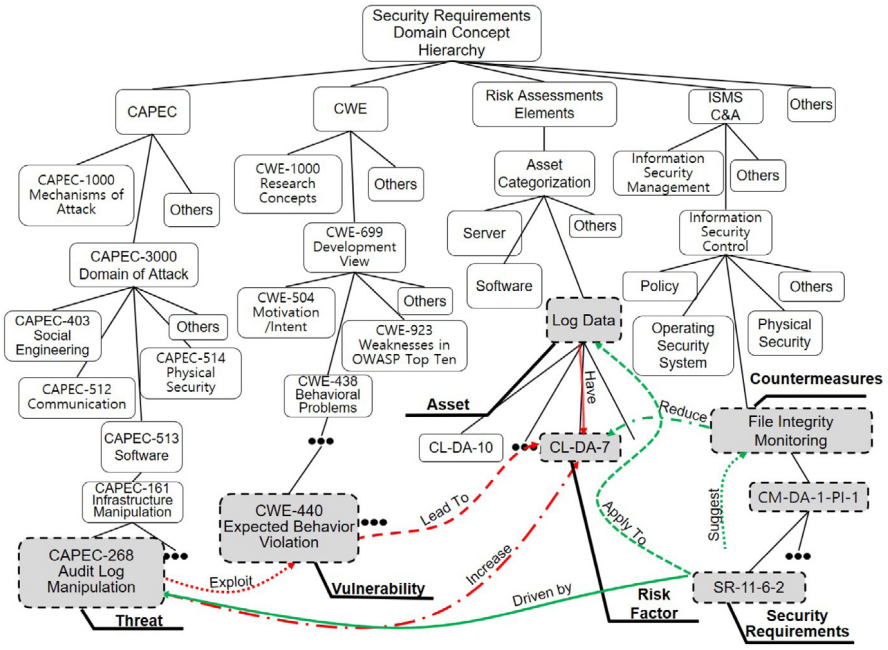


Fig. 8. Security requirements concept hierarchy and relationships between concepts.

Table 5
Security requirements concepts and attributes.

Concepts	Attributes
Assets	Description, Related Stakeholders, Security Goal, Concern List
Security Goal	Security Goal
Security Requirements	Description, Recommendations
Threat	Description, Attack Vectors, Malicious Goal
Vulnerability	Platform, CVE, CWE, Attack Vectors
Countermeasures	Description, Security Goal, Implementable countermeasures
Malicious Goal	Malicious Goal
Risk	Malicious Goal, Concern List

the concepts are classified and categorized shown in Fig. 7. Fig. 7 shows that various attributes from documents in the physical layer are used to consider security requirements.

Building the Security Requirements Concept Hierarchy Model

This step builds a hierarchy model by categorizing all elements in the security requirements knowledge sources. All the elements in these sources are represented using the hierarchical model and the relationship between concepts based on the agreements among SMEs. After building the hierarchy model, the elements in this model are given an identification number in order to manage the concepts, attributes, and relationships. If the identification number is already given, as in CAPEC, CWE, and CVE, we follow the original numbering system. Otherwise, we give a customized identification number based on the element type. For example, a server access control in the countermeasure can be granted ID, CM_DA_1_PI_1. CM means countermeasure, DA_1 refers to Log Data, and PI_1 is Preventing Integrity.

Specifying the General Purpose Knowledge Base

This step creates the GPK based on artifacts from previous steps. The hierarchy model is implemented with the concepts and relationships using an ontology tool, such as Protégé (Horridge, 2011). All the elements in GPK are specified based on the knowledge source, and stored in the database using the identification number.

Performing Activities of Phase 1 for the case scenario

The preparation step is the same as that shown in the previous section. The second step, building a security requirements concept hierarchy model, focuses on the File Integrity Monitoring

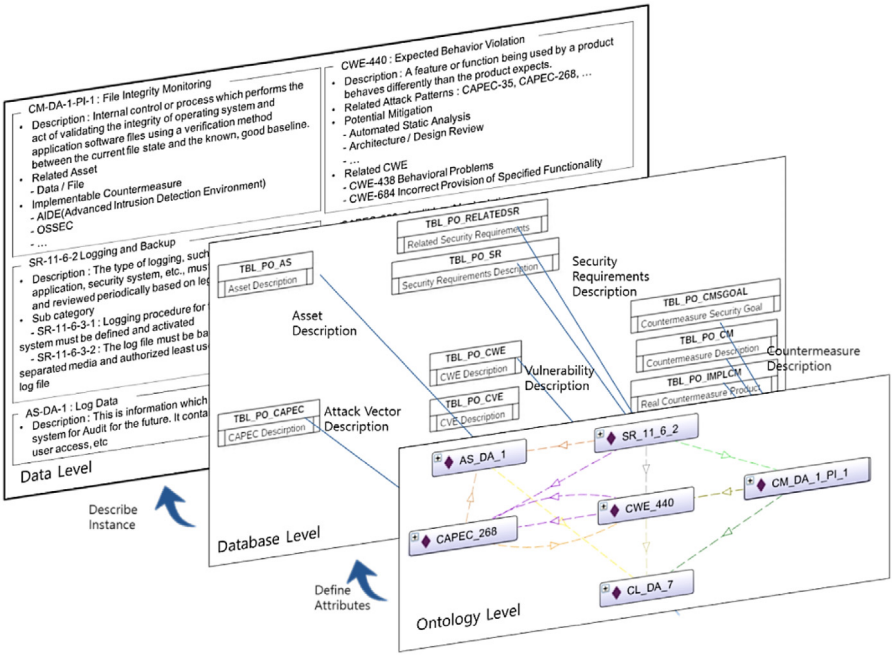


Fig. 9. An example structure and specifications of the Build GPK phase.

(FIM) against Hacktool.eventlog. As shown in Fig. 8, FIM is a process or control system used to monitor the file integrity. By using FIM and other related concepts, the relationship model is created. FIM is marked using the ID number, CM-DA-1-MI-1, as a member of Monitoring Integrity (MI) for Log Data (DA-1). As a countermeasure, OSSEC and Advanced Intrusion Detection Environment (AIDE) are implemented. This countermeasure is mapped to the Log and Backup system in the ISMS certification for security requirements, SR-11-6-2. Related assets such as the server and data are applied to the relationship with the requirements. Even though various weaknesses are mitigated by this countermeasure, we specify the expected behavior violation for this case study, CWE-440. The CAPEC element, which exploit this weakness is the audit log manipulation, CAPEC-268. This countermeasure reduces the risk of mistreating data, CL-DA-7. After the requirements extraction step, the GPK is created as shown in Fig. 9, which shows the partial knowledge base for this case study. Fig. 9 is an example structure and specifications of the Build GPK phase. In the case of SR_11_6_2 on the ontology level, it can be implemented to TBL_PO_RELATEDSR and TBL_PO_SR in GSK related attributes, and then described by instances.

3.2.3. Phase 2: Building the domain specific knowledge base

This phase builds the DSK with domain-specific information and creates the PDO by integrating GPK and DSK (Table 3). The domain working group performs phase 2 and works with the analysis of the architecture and threat information. We have developed an automation tool for supporting phases 2 and 3.

Building Architecture Information

This step collects all data and information related to the architecture, and elicits related concepts through a system or business analysis of the domain or the enterprise. This research assumes the framework already has the enterprise architecture. The related stakeholder, asset, platform, and operating countermeasure information is identified using domain architectures, and utilized in the risk assessment to recommend security requirements.

When identifying the asset information, it is necessary to set the security goal of the asset. Table 6 shows an example of setting the asset security goal with the log data. Firstly, the security attribute, malicious goal, and countermeasure type of the related asset need to be identified. Security attributes consist of Confidentiality (C), Integrity (I), Availability (A). Malicious goals can be marked with Exposure (E) Integrity (I), Modification (M). Types of countermeasure include fault tolerance mechanisms (Butler, 2008), monitoring, prevention, and recovery. Using this categorization, the importance of the countermeasure type is marked as C (Critical) and N (Not Critical). C represents the highest priority, which means that the marked countermeasure type is required, and N represents the lowest priority, which means that the marked countermeasure type is not required. Then, the result of marking the security goal per countermeasure type is calculated as RM (Required Monitoring), RP (Required Prevention), and RR (Required Recovery). In particular, C means that all countermeasure types are required.

Table 6
Asset Security Goal Calculating Matrix.

Security Attribute	Malicious Goal	Asset			Security Goal
		Monitoring	Prevention	Recovery	
C	E	N	C	N	RP
I	M	C	C	C	C
A	D	C	N	C	RMRR

Threat Analysis

SPIC Framework analyzes threats using Open OME Tools with the i* frame- work (Yu, 1997). The advantage of i* framework is that the actors, goals, tasks, and resources are easily identifiable. It enables the analysis of a threat through the requirements engineering approach, and can easily be applied to software engineering using the Tropos methodology (Bresciani et al., 2004). By utilizing this advantage, the malicious actors, malicious goals, tasks, and resources used in achieving the malicious goal are modeled. We customize several notations in order to model the threat for the SPIC framework using the Attack Goal Based Model (AGBM) by Kim and Lee (2015). Finally, the identified elements are mapped to the DSK as the threat information using customized notation. An example of the threat analysis is shown in the case study.

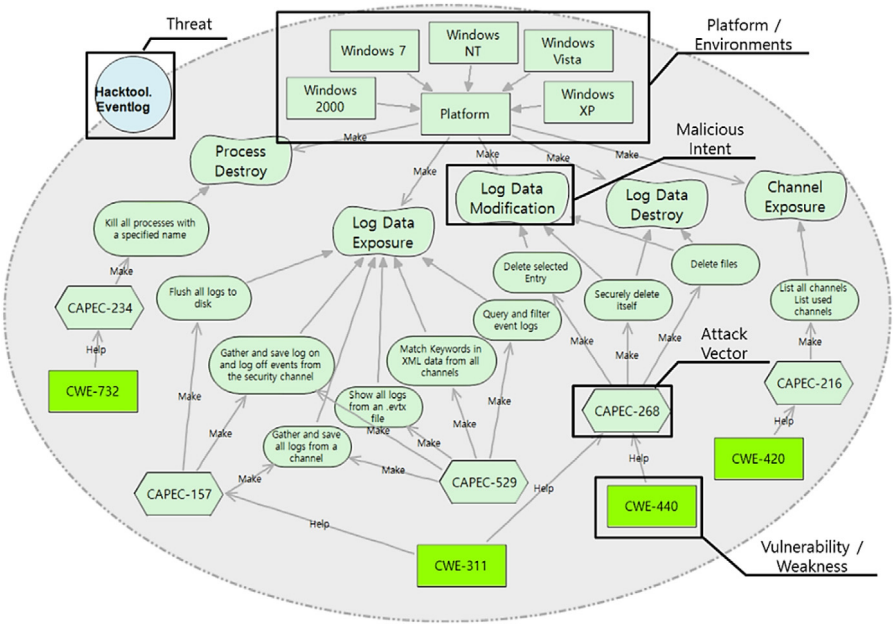


Fig. 10. Results of the threat analysis (Hacktool.Eventlog).

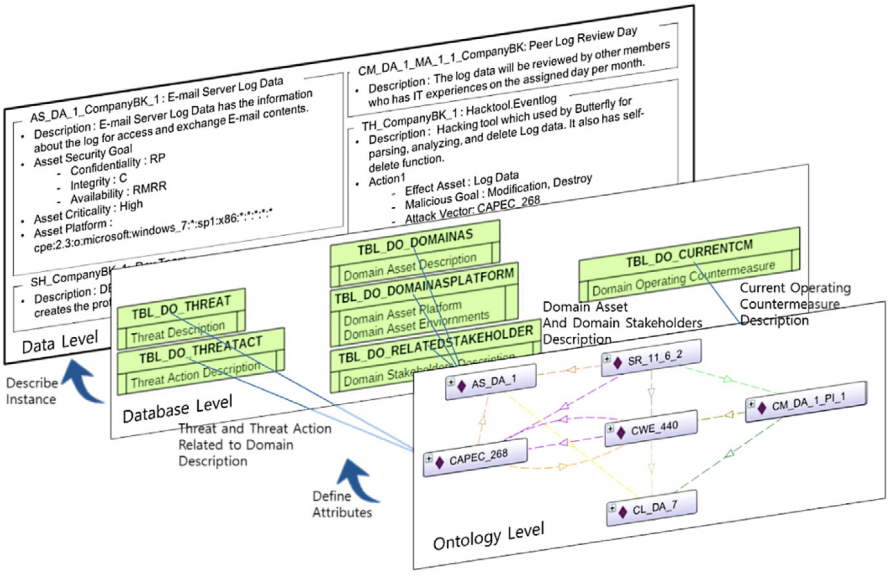


Fig. 11. An example structure and specification of Building DSK.

Specify Domain Specific Knowledge Base

The concepts and information identified from the architecture information and threat analysis are specified based on the GPK information. The specified information in DSK is also given an identification number to manage the information and build PDO by integrating with GPK. It is noteworthy that the PDO is different in each domain or enterprise due to the DSK.

Performing activities of Phase 2 for the case scenario

Firstly, the architecture information of the BK Company is identified using environments specifications. The team information in the network topology provides the stakeholder information for the BK Company. Then, the asset information is identified with the platform and security goal. Finally, the current operational countermeasure in the BK Company, which is shown in Table 4, is presented into the SPIC framework with an identification number.

After inserting the architecture information, the threat analysis is conducted with the i* framework. Fig. 10 is the diagram of the threat analysis (Hacktool.Eventlog). As shown in Fig. 10, malicious tasks by Hacktool.Eventlog are identified. The malicious goal of the threat is then identified based on the tasks and related assets. In this case study, we set Modification of the Log Data as the malicious goal, which is achieved by Delete Selected Entry. This task is implemented using the attack vector, CAPEC-268 (Audit Log Manipulation), and it exploits the weakness, CWE-440 (Audit Log Manipulation). This tool is operated on a Windows server matching the specific version. The entire diagram element shown in Fig. 10 is integrated into the DSK in Fig. 11. Fig. 11 is an example structure and specification of Building DSK. In the ontology level of PDO, there are the concepts such as AS_DA_1, CAPEC_268, CWE_440, SR_11_6_2, CL_DA_7, and CM_DA_1_PI_1 and the relation of these concepts. These concepts are mapped to entities in the database

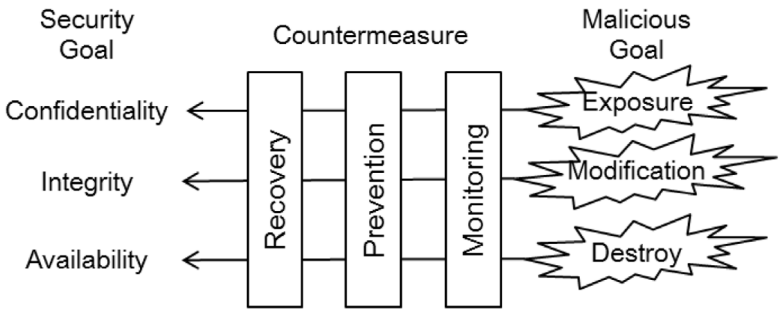


Fig. 12. Risk assessment reference model.

No	Description	Question	Metric	Measure	Result
1	Platform information? (No / Same or Similar)	Is the platform of system or asset the same as the required platform of vulnerability?	• System or platform information • Vulnerability required platform information	Same or similar	Risk Exist / Step 3
				No	Step 2
2	Security Configuration? (No / Yes)	Does the security configuration information have inherent weakness in the platform or security service?	• Security Configuration information	Yes	Risk Exist / Step3
				No	No Risk
3	Define Asset Information	Define the Asset information in the risk Assessment Template.	• Asset Information	Asset Info / ID	Next Step
			• Asset Security Goal	C(Critical) R(Required) N(No required)	
4	Define Threat and Vulnerability Information	Define the threat and Vulnerability and malicious intent.	• Threat Name	Threat / ID	Next Step
			• Vulnerability Name	Vulnerability / ID	
			• Malicious Intent - E(Exposure) - M(Modification) - D(Destroy)	Yes / No	
5	Define the Current Countermeasure	Define the current countermeasure(CM) based on the system architecture.	• Monitoring CM - C / I / A	Yes / Weak / No	Next Step
			• Protection CM - C / I / A	Yes / Weak / No	
			• Recovery CM - C / I / A	Yes / Weak / No	
6	Assess Risk	Are there any achieved malicious intentions based on the Risk Assessment Template?	• Metric - Asset Security Goal - Current CM Info - Malicious Goal • Risk Factor - E / M / D	Yes / No	Next Step
7	Define recommended Countermeasures	Recommend Countermeasures based on the Risk Factor	• Risk Factor - E / M / D	Counter-measures type	Finish

(a) Risk Assessment Process based on PDO

Step 3 Step 4 Step 5 Step 6 Step 7

Asset	Asset Sec. Goal			Threat	Vulnerability	Malicious Goal			Current Countermeasures									Risk Factor			Recommen-dations
									C			I			A						
	C	I	A			E	M	D	M	P	R	M	P	R	M	P	R	E	M	D	

(b) Risk Assessment Template

Fig. 13. Goal-oriented risk assessment processes and templates.

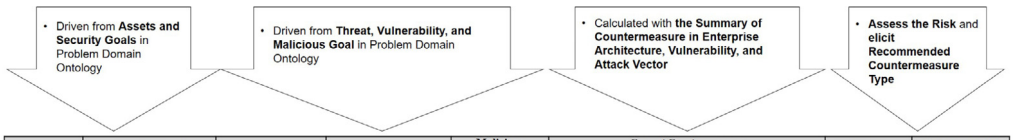
level of PDO. For example, AS_DA_1 in the ontology level are mapped to TBL_DO_DOMAINS, TBL_DO_DOMAINSPLATFORM, and TBL_DO_RELATEDSTAKEHOLDER with a DSK related attributes in a database level. As shown in Fig. 11, the architecture and threat information consist of the PDO based on the GPK and DSK.

3.2.4. Phase 3: Recommending security requirements

As the last phase of the SPIC framework, this phase performs risk assessment based on the asset information and threat analysis (Table 3). It also recommends the security requirements specification based on the result of the risk assessment.

Risk Assessments

We propose a goal-based risk assessment using the security attribute triad for the risk assessment reference model. Fig. 12 shows the security goal with the malicious goal, Confidentiality with Exposure, Integrity with Modification, and Availability with Destroy. It means the exposure hurts confidentiality; integrity is hurt by the modification of an asset; and availability



Asset	Security Goal			Threat	Attack Vector / Vulnerability	Malicious Goal			Current Countermeasures									Risk Factor			Recommended Countermeasure Type
									Confidentiality			Integrity			Availability						
	C	I	A			E	M	D	M	P	R	M	P	R	M	P	R	E	M	D	
Log Data	RP	C	RMRR	Hacktool.Eventlog	CAPEC-268/CWE-440		V	V	W	W	N	N	N	N	W	W	Y		V	V	MI, PI, RI, MA
Log Data	RP	C	RMRR	Hacktool.Eventlog	CAPEC-157/CWE-331	V			W	W	N	N	N	N	W	W	Y	V			PC
Process	N	C	C	Hacktool.Eventlog	CAPEC-234/CWE-732			V	N	N	N	Y	Y	Y	Y	Y	Y				
Channel	C	RMRP	RPRR	Hacktool.Eventlog	CAPEC-216/CWE-420	V			Y	Y	N	N	N	N	Y	Y	W	V			RC
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Fig. 14. Case study: Risk assessment.

is lost by the destruction of the asset. These malicious goals are achieved by malicious actors through security intrusion. In this reference model, the countermeasure is used to control the malicious intention and enhance the security goal. We define the countermeasure type based on the fault tolerance mechanisms proposed by Ricky Butler (Butler, 2008), Monitoring, Prevention, and Recovery. Risk assessments are performed based on these components. The malicious goal of the threat is achieved when the required countermeasure of each asset security goal is not operated. On the other hand, the security goal of the asset is achieved when the required countermeasure is operated properly. In addition, the risk assessment is performed with two main criteria, namely potential loss and probability of the risk. A High probability of the risk means that the required countermeasure of each security goal is not available. On the other hand, a Low probability of the risk means that the required countermeasure of each security goal is available. The potential loss of the asset is determined by the asset criticality when the malicious goal is achieved. The result of the risk assessments recommends the required countermeasure type based on this reference model.

The goal oriented risk assessment process based on the reference model is shown in Fig. 13(a). The steps 1 and 2 are used to determine Risk Exist using the system environments and configuration information. The information about the system environments, including the platform or version, identifies risk probability. Vulnerability by weak security configurations is considered in determining the probability of risk occurrence, such as having a weak password or encryption mechanism. When the result is Risk Exist, risk assessment is performed according to the template shown in Fig. 13(b). The outcome of the template is the recommended countermeasure type based on the risk assessment of the security and malicious goal.

Firstly, step 3 is performed to define the asset security goal from the goal-calculating matrix shown in Table 5. Step 4 then identifies the threat, malicious goal, and vulnerability using the attack goal-based model from the threat analysis. The malicious goal of the threat is defined using E (Exposure), M (Modification), and D (Destruction). For example, E means that the related asset is exposed to malicious actors without an appropriate required countermeasure. Step 5 identifies the current operating countermeasures in the domain or the enterprise based on the information from previous phases. Each countermeasure type is marked with Y (Yes) or N (No). When the related security service is not operated properly, the related countermeasure type is marked with W (Weak). Then, step 6 decides the risk factors by comparing the required countermeasure of the asset with the operating countermeasure. For example, if the availability of the asset security goal is marked as C, all related countermeasure types are satisfied. When the current countermeasure is estimated as W, the destruction, which is symmetric with the availability, is marked as Y. Finally, step 7 provides the recommended countermeasure based on the asset security goal.

Elicit Recommended Countermeasure

This step elicits the recommended countermeasure based on the result of the risk assessment. Implementable countermeasures are obtained from the recommended countermeasure type using the PDO. After the countermeasure is selected from the recommended countermeasure type, all the concepts for specifying the security requirements are gathered.

Specify Security Requirements

This is the final step in specifying the security requirements. It consists of the security requirements diagram and the template with the related concepts, such as the asset, countermeasure, threat, etc. When extending the GPK with the agreements among SME groups, it is possible to provide more implementable security requirements using the similarity comparison.

Performing activities of Phase 3 for the case scenario

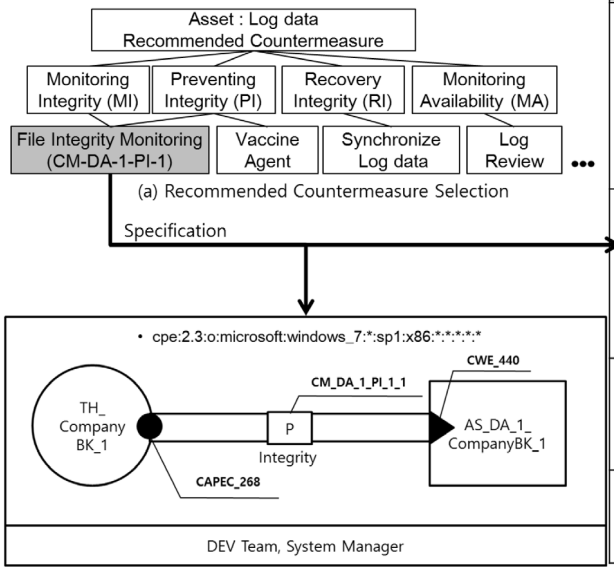
In this case study, it is assumed the platform information is the same as that corresponding to the threat. With the result of Risk Exist, the risk assessment process is conducted with the template and PDO, as demonstrated in Fig. 13. Since the threat in the case study affects the Log Data, the security goal of integrity is marked C, as shown in Fig. 14. After that, the threat information and current operational countermeasures are identified in steps 4 and 5. Step 6 provides the risk factors based on the current countermeasure, the security goal, and the malicious goal. For example, the security goal related to integrity requires all types of countermeasures. Since there is no current operational countermeasure related to integrity in the BK Company, all the countermeasures are marked as N, and *Modification* and *Destruction*, as the malicious goal against the log data, can be achieved. Based on the result of the risk factor, the recommended countermeasure type for the BK Company is provided.

When completing the risk assessment, the countermeasure is selected based on the result of the risk assessment, i.e. the recommended countermeasure type. As shown in Fig. 15(a), the categorized implementable countermeasures are presented according to the security goal in Log Data. In this case, we select the FIM for Preventing Integrity and specify the related security requirements with related concepts from the PDO. Fig. 15(b) shows a diagram of the security requirements. In this diagram, risk components are identified such as TH_CompanyBK_1 as a threat, AS_DA_1_CompanyBK_1 as an asset, CAPEC_268 as an attack vector, CM_DA_1_PI_1_1 as a countermeasure, DEV Team, System Manager as a stakeholder and so on. Fig. 15(c) shows a template for the security requirements specification in this case study.

4. Support tools

4.1. Introduction to SPICA

Security Requirements Physical, Information-modeling, and Cognitive Framework Application (SPICA) is a prototype tool developed to apply the SPIC framework with automation. The



Category	Attribute	ID	Content
Security Requirements	Name	SR-11-6-2	• Logging and Backup
	Description		• The type of logging, such as information system, application, security system, etc., must be defined, preserved, and reviewed periodically based on legal compliance
	Source		• KISA ISMS Certification Requirements
	Sub-Requirements	SR-11-6-2-1	• Logging procedure for the critical information system must be defined and activated
		SR-11-6-2-2	• The log file must be backed up using separated media and authorized least users for accessing the log file
Counter-measure	Type	C-DA-1-PI-1	• Monitoring Integrity / Preventing Integrity
	Name		• File Integrity Monitoring
	Description		• Internal control or process which performs the act of validating the integrity of operating system and application files using a verification method between the current file state and the known, good baseline.
	Implementable Counter-measure		• AIDE(Advanced Intrusion Detection Environment) • OSSEC • …
Asset	Asset Name	AS_DA_1_ CompanyBK_1	• Log Data
	Related Stakeholders		• System Manager / Security Staff / …
	Asset Security Goal	AS-DA-1-SG-C	• Require Protection(RP)
		AS-DA-1-SG-I	• Critical(C)
		AS-DA-1-SG-A	• Require Monitoring / Require Recovery(RMRP)
Risk Factor	Threat	TH_ CompanyBK_1	• Hacktool.Eventlog
	Achieved Malicious Goal		• Modification Log Data • Destroy Log Data
	Vulnerability	CWE-440	• Expected Behavior Violation
	Attack Vector	CAPEC-268	• Audit Log Manipulation

(c) Security Requirements Specification

Fig. 15. Case study: Recommended security requirements specification.

```
SELECT ?SR {?SR SROnt:suggest SROnt:CM,
                SROnt:apply_To SROnt:AS,
                SROnt:prevent SROnt:CWE,
                SROnt:driven_By SROnt:CAPEC}
```

Fig. 16. An example of SPARQL query code for the SPIC framework.

automation of phase 1 is not implemented in the SPICA prototype, because it is assumed that the GPK is already built. The services provided by SPICA are related to phases 2 and 3, namely build a domain-specific knowledge base and recommend security requirements. The concept and relationship in GPK from phase 1 are implemented using the ontology and the database.

SPICA is developed with a server–client structure. The client side is implemented using Hyper Text Markup Language (HTML) and JavaScript. The server side is implemented using the Spring Framework and Apache server. The ontology part is implemented using Protégé 4.3 with the Jena API for the server side. In order to use the inference of the ontology, we use a SPARQL (Prud'hommeaux E, 2008) query, which is shown in Fig. 16, to obtain the other concept related to the security requirements.

The "Building Architecture information" in phase 2 stores the architecture information, such as the stakeholder information, platform information, asset information, and current operating countermeasures, into the database. In the "Threat Analysis" step, the result of threat modeling with the i* framework is mapped to SPICA. Based on the PDO, SPICA assesses the risk and provides the result of the risk assessment. Then, the recommended security requirements are provided based on the result of the risk assessment. Fig. 17 shows an example of risk assessment in SPICA.

In the step 1 and 2, Risk Exist is determined using the system environments and configuration information. These results are shown in "Check Risk Existence" table. There are two Risk Exist rows. In step 3, the asset security goals are defined from the goal-calculating matrix. In step 4, the threat, malicious goal, and vulnerabilities are identified using the attack goal-based model from the threat analysis. The malicious goal of the threat is defined using E (Exposure), M (Modification), and D (Destruction). Each countermeasure type is marked with Y (Yes) or N (No). When the related security service is not operated properly, the related countermeasure type is marked with W (Weak). The current operating countermeasures are defined in step 5, and then risk factors are decided by comparing the required countermeasure of the asset with the operating countermeasure in step 6.

4.2. Usage of SPICA

The functions of the SPICA prototype are to store the domain-specific knowledge based on GPK, to assess the risk, and to specify the recommended security requirements using the PDO. The advantage of SPICA is that it enables automation of the SPIC framework and standardization of information on a large scale. The future SPICA is aimed at building the GPK phase with ontology information, and more sophisticated recommended-security-requirements with a similarity comparison of the threat and zero-day vulnerability.

4.3. Case study with SPICA

This section demonstrates the result of the SPIC framework for the case study scenario using SPICA. Fig. 14 shows the result of the risk assessment based on the related concepts, information, and data, which is the same scenario as that of the case study. Fig. 18 is the security requirements specification using SPICA for the case study. This specification includes security requirements, countermeasure, asset, and risk factor. Each component has attributes, ID numbers, and contents. These can support understanding of the security requirements for the case study scenario. Based on this application, we understand the security environment of the domain with the standardized information and recommend the security requirements with the specification automatically.

5. Evaluation

This paper validates the SPIC framework based on the How and Why analysis approach described in Lee and Rine (2004). Our main research question is:

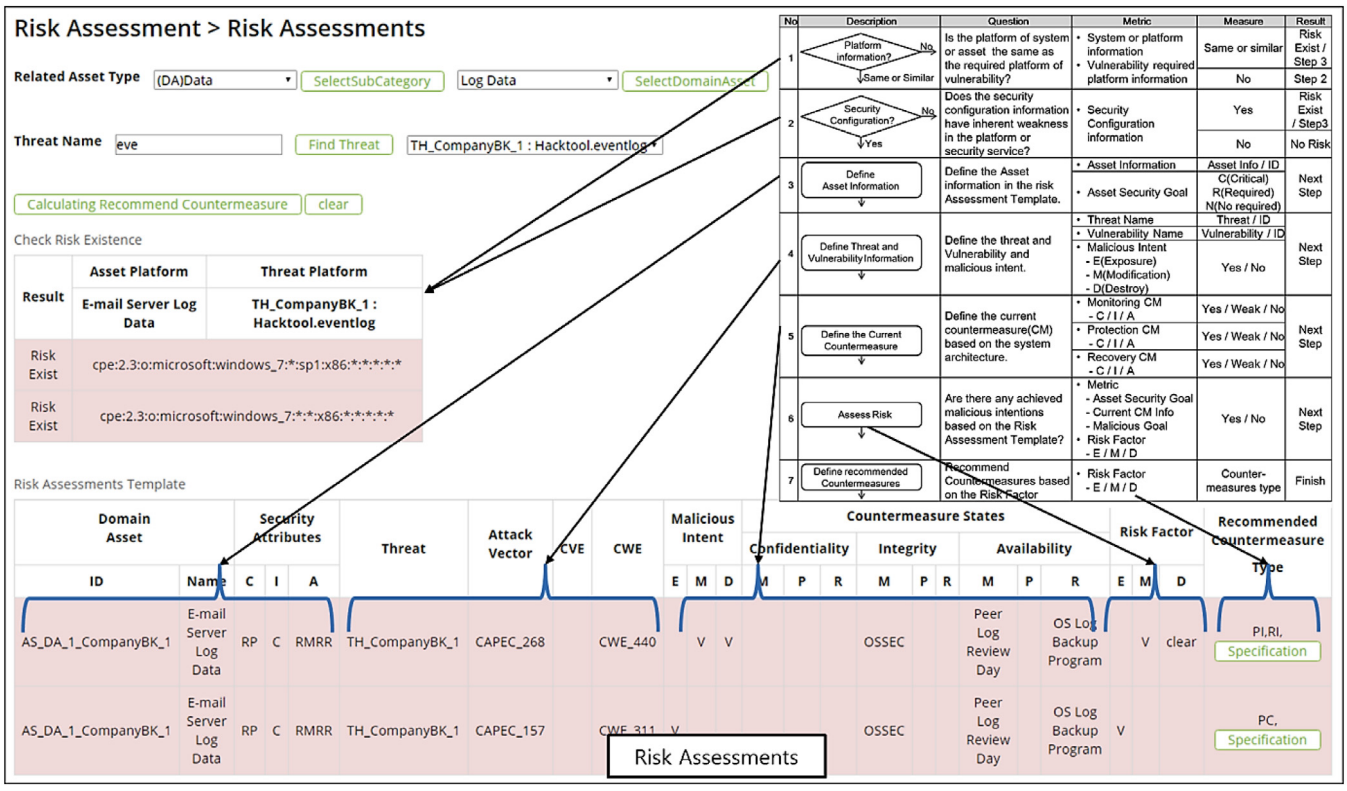


Fig. 17. Risk Assessment Process in SPIC framework and SPICA (risk assessment example).

Table 7
Concept and related knowledge sources for understanding security environments.

Concept	Knowledge Source
Threat	Response Report, CAPEC
Vulnerability	CVE, CWE
Stakeholders	Enterprise Architecture
Domain Knowledge	Enterprise Architecture, System Architecture, Platform Information
Asset	Enterprise Architecture, System Architecture, Asset Classification
Security Requirements	ISMS C and A Item, Law Information
Risk	ISMS Risk Management
Countermeasure	Countermeasure Classification
Security Goal	CIA, Other Requirements

RQ: "How and why does this research help understand critical security related issues of complex socio-technical systems?"

For this, we define the general proposition (GP) of the research for the analytical validation in supporting the research goal.

"The SPIC framework helps achieve the research questions, because the SPIC framework provides the guidance to organize the scattered STS knowledge and models, and to perform the requirements engineering process efficiently by understanding and recommending the security requirements with the specification".

In order to achieve the GP, we set three specific propositions (SPs). We validate each SP based on the evidence from the case study.

SP1. The SPIC framework helps users organize and integrate various related knowledge.

– Set of documents or knowledge sources: comparing the number of utilized knowledge sources with other modeling methodologies using a fixed set of knowledge sources.

– Set of models: comparing the utilized modeling approaches with other modeling methodologies.

SP2. The SPIC framework helps users understand the security environment of the targeted STS from security concepts and relationships.

– Elements to understand security requirements: comparing the elements with those of other modeling methodologies.

SP3. The SPIC framework helps users recommend the specified security requirements with automated security requirements engineering methodologies.

– Automation item: providing the results of the specified security requirements using a tool that supports security requirements engineering.

5.1. Evidence collection from the case study result

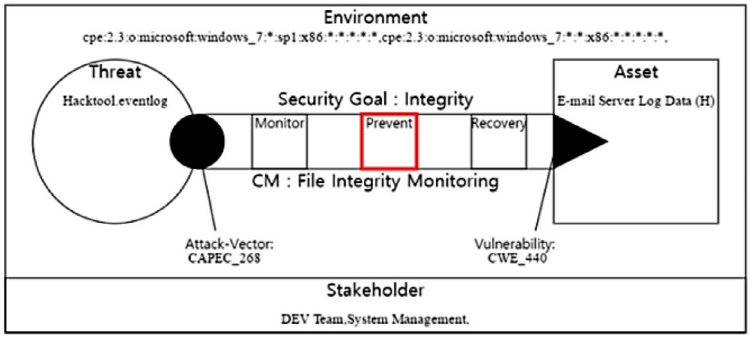
SP1 and SP2:

In order to validate SP1 and SP2, the set of knowledge sources and the approach for the modeling methods, which is shown in the analysis section, are compared with the other methodologies presented in Section 2. We assume that the security

Risk Assessment > Recommending Security Requirements with Contexts

PI : File Integrity Monitoring | Recommend Security Requirements

Recommended Security Requirements and Context



category	Attributes	ID Number	Contents
Security Requirements	name	SR_11_6_2	Logging and Backup
	Description	SR_11_6_2	The type of logging, such as information system, application, security system, etc., must be defined, preserved, and reviewed periodically based on legal compliance
	Related Requirements	SR_11_6	Log Management and Monitoring
		SR_11_6_2_1	Logging and Backup1
		SR_11_6_2_2	Logging and Backup2
Countermeasure	name	CM_DA_1_PI_1	File Integrity Monitoring
	Description	CM_DA_1_PI_1	Internal control or process which performs the act of validating the integrity of operating system and application software files using a verification method between the current file state and the known, good baseline.
	Implementable Countermeasure	CM_DA_1_PI_1_1	Advanced Intrusion Detection Enviornment
		CM_DA_1_PI_1_2	OSSEC
Asset	Type	AS_DA_1	Log Data
	Protected Domain Asset	AS_DA_1_CompanyBK_1	E-mail Server Log Data (H)
	Security Goal	AS_DA_1_CompanyBK_1_Confidentiality	RP
		AS_DA_1_CompanyBK_1_Integrity	C
		AS_DA_1_CompanyBK_1_Availability	RMRR
	Related Stakeholders	SH_CompanyBK_1	DEV Team
		SH_CompanyBK_4	System Management
Risk Factor	Threat	TH_CompanyBK_1	Hacktool.eventlog
	Threat Description		hacking tool which used by Butterfly for parsing, analyzing, and delete Log data. It also has self-delete function.
	Threat Activity		* Delete Selected Entry * Securely Deleted itself * Delete Files
	Malicious Goal		Modification, Destroy
	Threat Attack Vector	CAPEC_268	Audit Log Manipulation
	Related Vulnerability		
	Related Weakness	CWE_440	Expected Behavior Violation

Fig. 18. Security requirements specification using SPICA for the case study.

Table 8
Comparison of the represented security requirements elements and knowledge sources with the modeling approaches of other modeling methodologies.

Methodology	SR Elements	Knowledge Sources		Modeling Approach
		Count	Set	
Proposed Framework	SG, TH, VU, RI, EN, AS, SR, ST, CM	14	Response Report, CAPEC, CVE, CWE, Enterprise Architecture, System Architecture, Asset Classification, Platform Information, ISMS C and A Process, Law Information, ISMS Risk Management, Countermeasure Classification, CIA, Other Requirements	Goal-based, CC, Risk Assessment, Multilateral
MSRA: Gürses et al. (2006)	AS, ST, SG, EN	6	Enterprise Architecture, System Architecture, Asset Classification, CIA, Other Requirements, Platform Information	Multilateral
SQUARE Mead and Stehney (2005)	ST, TH, SG, RI	6	Enterprise Architecture, Response Report, CAPEC, CIA, Other Requirements, ISMS Risk Management	Multilateral, UML-based.
Misuse case Sindre and Opdahl (2001)	SG, TH, VU, RI, ST, AS	10	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, ISMS Risk Management, Enterprise Architecture, System Architecture, Asset Classification	UML-based
Secure UML Lodderstedt et al. (2002)	ST, EN	3	Enterprise Architecture, System Architecture, Platform Information	UML-based
UMLsec Jürjens (2005)	SG, TH, VU, RI, ST, EN,	10	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, ISMS Risk Management, Enterprise Architecture, System Architecture, Platform Information	UML-based
KAOS anti-model Van Lamsweerde (2004)	SG, TH, VU, ST, EN, AS	10	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, Enterprise Architecture, System Architecture, Asset Classification, Platform Information	Goal-based
Secure Tropos Mouratidis and Giorgini (2007)	SG, TH, VU, RI, ST, SR	10	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, ISMS Risk Management, Enterprise Architecture, ISMS C and A Item, Law Information	Goal-based
GBRAM Antón and Earp (2001)	SG, TH, VU, RI, ST, AS	10	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, ISMS Risk Management, Enterprise Architecture, System Architecture, Asset Classification	Goal-based
STS FW Paja et al. (2015)	SG, ST, EN	5	CIA, Other Requirements, Enterprise Architecture, System Architecture, Platform Information	Goal-based
Three layered FW Li and Horkoff (2014)	SG, TH, VU, AS	9	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, Enterprise Architecture, System Architecture, Asset Classification	Goal-based
CORAS Den Braber et al. (2006)	TH, VU, RI, EN, AS, CM	10	Response Report, CAPEC, CVE, CWE, Enterprise Architecture, System Architecture, Platform Information, Asset Classification, ISMS Risk Management, Countermeasure Classification	Risk Assessments
Tropos goal-risk FW Asnar et al. (2007)	SG, TH, RI, ST	6	CIA, Other Requirements, Response Report, CAPEC, ISMS Risk Management, Enterprise Architecture	Risk Assess ments,Goal-based
CC (Common Criteria, 2012)	SR, TH, VU, RI, SG, EN, AS, CM	14	Response Report, CAPEC, CVE, CWE, System Architecture, Asset Classification, Platform Information, ISMS C and A Process, Law Information, ISMS Risk Management, Countermeasure Classification, CIA, Other Requirements	CC
SREP Mellado et al. (2006)	SG, TH, VU, RI, EN, AS	12	CIA, Other Requirements, Response Report, CAPEC, CVE, CWE, ISMS Risk Management, Enterprise Architecture, System Architecture, Platform Information, Asset Classification	CC, Risk Assessment, Multilateral, UML-based

requirements are generated only from the fixed set of knowledge sources with each modeling methodology as shown in Table 7. For example, threats are obtained from response reports of many incidents and accidents, and various vulnerabilities of assets can be acquired from CVE (Common Vulnerabilities and Exposures) and CWE (Common Weakness Enumeration), and environment information can be acquired from enterprise architecture, system architecture, and platform information. In addition, the duplicated knowledge sources are ignored. As the methodology uses more knowledge sources in this analysis, the methodology provides more reliable and richer security context-awareness with systematical knowledge organization. Based on this assumption, we compare with other modeling methodologies. The results of this comparison are shown in Table 8. The proposed framework can understand and recommend security requirements based on nine elements such as the SG (Security Goal), TH (Threat), VU (Vulnerability), RI (Risk), EN (Environment), AS (Asset), SR (Security Requirement), ST (Stakeholder), and CM (Countermeasure). These elements can originate from the CIA & Other Requirements, Response Report & CAPEC, CVE & CWE, ISMS Risk Management, Platform Information, System Architecture & Asset Classification, ISMS C&A Item & Law Information, Enterprise Architecture and Countermeasure Classification, respectively. This framework includes more security requirement elements than other modeling

Table 9
Automation elements of the SPICA prototype.

Category	Non-Automated Step	Automated Step
Proposed Framework	* Building General Purpose Knowledge Base * Modeling the Architecture Information * Threat Analysis	* Building Domain Specific Knowledge Base using GPK * Risk Assessment (Fig. 17) * Recommend Countermeasure Type (Fig. 15) * Specify Security Requirements (Fig. 18)

methodologies such as MSRA, SQUARE, Misuse case, etc., because it accepts a variety of approaches including those that are Goal-based, CC, Risk Assessment, and Multilateral.

The results show the set of knowledge sources are dependent on the elements of the modeling methodology for understanding the security context-awareness. The proposed framework has the advantages of integrating various modeling approaches and understanding the security context-awareness related to the security requirements.

SP3:

SP3 is validated using SPICA artifacts and the table listing the automated item set, which is shown in Table 9. SPICA provides automated steps such as building the Domain Specific Knowledge Base using GPK, assessing risk, recommending countermeasure type and specifying security requirements. Even though the modeling methodologies described in Section 2 help provide the well-represented security requirements by SME, the SPIC framework has the advantage of an automated specification of recommended security requirements with the related concepts.

5.2. Analysis of the validation and contribution

As shown in the previous sub-section, the SPIC framework supports the GP and SPs. Table 8 shows the set of knowledge sources according to the modeling concept and the modeling approach of each modeling methodology described in Section 2. This result shows the SPIC framework enhances the security context-awareness using various concepts by integrating various modeling methodologies, and recommends reliable security requirements by fully understanding the security environments of the target STS. As a result, SP1 and SP2 are validated based on this evidence. SP3 is related to the automation of the SPIC framework for understanding and recommending the security requirements. The SPICA prototype supports the automation of phases 2 and 3 by specifying the security requirements with related concepts. By using the SPICA, the domain working group recommends the security requirements automatically with the PDO. Based on the case study and analytical validation, it is confirmed the SPIC framework supports the SPs and GP.

6. Conclusion

Ensuring the security of complex socio-technical systems is a great challenge as we currently lack a method for the systematic acquisition of the scattered knowledge related to the design, development and execution of STS. In this paper, we attempt to address security issues by providing guidelines to organize the scattered source of knowledge. We propose a cognitive three-layered framework SPIC integrating various modeling methodologies and knowledge sources related to security. This framework helps in understanding and recommending security requirements using threat analyses and goal-based risk assessments. The impact of this research is that it provides a methodology to understand and recommend security requirements with a systematic organization of knowledge using the PDO. We conducted a case study to demonstrate the highly applicable nature of this SPIC framework involving a real threat. We finally validate the framework by collecting evidence supporting the GP and SPs. Moreover, the tool support helps the domain working group understand and recommend the explicit security requirement automatically.

A scalability problem occurs, however, when handling large amounts of knowledge due to the inherent limitation of ontology, which results in the problem of ontology maintenance. Moreover, if the knowledge base is not built based on the expertise of SMEs, an incorrect knowledge base could lead to defects in security requirements. Even though the risk assessments in the SPIC framework provide a qualitative goal-oriented analysis, it still needs to conduct a quantitative assessment using a complex case. In addition, it is necessary to standardize knowledge, including definition and categorization, in order to obtain an interoperable framework. Lastly, since the analytical approach is used for validating the SPIC framework, it is essential to further conduct an empirical study with a statistical approach for our future work. We can also further explore how the SPIC framework can be integrated into an Information Security Management System (ISMS), to reduce organizational risks and ensure business continuity. It is essential to evaluate how the final recommended security requirements contribute to the improvement of information security in STS.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1F1A1075605). The authors would like to thank anonymous reviewers for their valuable comments. We are also immensely grateful to Sangeeta Dey and Sihn Hye Park for their help on the manuscript.

References

Alberts, C., Dorofee, A., Stevens, J., Woody, C., 2003. Introduction to the Octave Approach. Pittsburgh. Tech. Rep., Carnegie Mellon University, PA.

Antón, A.I., Earp, J.B., 2001. Strategies for developing policies and requirements for secure and private electronic commerce. In: E-Commerce Security and Privacy. Springer, pp. 67–86.

Asnar, Y., Giorgini, P., Massacci, F., Zannone, N., 2007. From trust to dependability through risk analysis. In: The Second International Conference on Availability, Reliability and Security. ARES'07, IEEE, pp. 19–26.

Bass, L., Clements, P., Kazman, R., 2012. Understanding quality attributes. In: Software Architecture in Practice. Pearson Education, pp. 63–78.

Baxter, G., Sommerville, I., 2011. Socio-technical systems: From design methods to systems engineering. Interact. Comput. 23 (1), 4–17.

Blank, R., Gallagher, P., 2012. Guide for conducting risk assessments. NIST Special Publication 800–30 Revision 1, 1–95.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J., 2004. Tropos: An agent-oriented software development methodology. Auton. Agents Multi-Agent Syst. 8 (3), 203–236.

Butler, R.W., 2008. A Primer on Architectural Level Fault Tolerance. Hampton: NASA Center for AeroSpace Information.

Caralli, R.A., Stevens, J.F., Young, L.R., Wilson, W.R., 2007. Introducing Octave Allegro: Improving the Information Security Risk Assessment Process. Tech. Rep., Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.

Collopy, P., 2012. A research agenda for the coming renaissance in systems engineering. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. p. 799.

Common Criteria, 2012. Part 1: Introduction and general model. In: Common Criteria for Information Technology Security Evaluation. pp. 43–62.
Den Braber, F., Brændeland, G., Dahl, H.E., Engan, I., Hogganvik, I., Lund, M., Solhaug, B., Stølen, K., Vraalsen, F., 2006. The CORAS Model-Based Method for Security Risk Analysis, Vol. 12. SINTEF, Oslo, pp. 15–32.
Elahi, G., 2009. Security Requirements Engineering: State of the Art and Practice and Challenges. Tech. Rep., Department of Computer Science, University of Toronto.
Elahi, G., Yu, E., 2007. A goal oriented approach for modeling and analyzing security trade-offs. In: International Conference on Conceptual Modeling. Springer, pp. 375–390.
Elahi, G., Yu, E., Li, T., Liu, L., 2011. Security requirements engineering in the wild: A survey of common practices. In: 2011 IEEE 35th Annual Computer Software and Applications Conference. IEEE, pp. 314–319.
Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H., 2010. A comparison of security requirements engineering methods. Requir. Eng. 15 (1), 7–40.
FireEye, Inc, 2019. FireEye Mandiant M-Trends 2019 report.
FireEye, Inc, 2020. FireEye Mandiant M-Trends 2020 report.
Ford, D., 2010. The OODA loop. http://www.danford.net/boyd/essence4.htm.
Grant, K.A., 2007. Tacit knowledge revisited–we can still learn from Polanyi. Electron. J. Knowl. Manage. 5 (2), 173–180.
Gürses, S., Berendt, B., Santen, T., 2006. Multilateral security requirements analysis for preserving privacy in ubiquitous environments. In: Proceedings of the UKDU Workshop. pp. 51–64.
Horridge, M., 2011. A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and CO-ODE Tools Edition 1.3. University of Manchester.
ISO/IEC, 2018. ISO/IEC 27005, Information technology -security techniques - information security risk management.
Jürjens, J., 2005. Secure Systems Development with UML. Springer Science & Business Media.
Kim, B.-J., Lee, S.-W., 2015. Conceptual framework for understanding security requirements: A preliminary study on stuxnet. In: Requirements Engineering in the Big Data Era. Springer, pp. 135–146.
Korea Internet and Security Agency, 2004. Guide to information security management system risk management. Tech. Rep., Korea Internet and Security Agency, Seoul.
Korea Internet and Security Agency, 2013. Information Security Management System Certification Policy Guidance. Tech. Rep., Korea Internet and Security Agency, Seoul.
Lee, S.W., Gandhi, R.A., 2005. Ontology-based active requirements engineering framework. In: 12th Asia-Pacific Software Engineering Conference. APSEC'05, IEEE, pp. 8–pp.
Lee, S.-W., Gandhi, R., Muthurajan, D., Yavagal, D., Ahn, G.-J., 2006. Building problem domain ontology from security requirements in regulatory documents. In: Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems. pp. 43–50.
Lee, S.W., Rine, D.C., 2004. Case study methodology designed research in software engineering methodology validation. In: SEKE. pp. 117–122.
Li, T., Horkoff, J., 2014. Dealing with security requirements for socio-technical systems: A holistic approach. In: International Conference on Advanced Information Systems Engineering. Springer, pp. 285–300.
Lodderstedt, T., Basin, D., Doser, J., 2002. SecureUML: A UML-based modeling language for model-driven security. In: International Conference on the Unified Modeling Language. Springer, pp. 426–441.
Mead, N.R., Stehney, T., 2005. Security quality requirements engineering (SQUARE) methodology. ACM SIGSOFT Softw. Eng. Notes 30 (4), 1–7.
Mellado, D., Fernández-Medina, E., Piattini, M., 2006. Applying a security requirements engineering process. In: European Symposium on Research in Computer Security. Springer, pp. 192–206.
Ministry of Government Legislation, 2016. National law information center. http://www.law.go.kr/eng/engMain.do.
MITRE Corporation, 2014. Common platform enumeration. http://cpe.mitre.org.
MITRE corporation, 2020. Common weakness enumeration(CWE). http://cwe.mitre.org.
MITRE Corporation, 2020a. Common attack pattern enumeration and classification(CAPEC). http://capec.mitre.org.
MITRE Corporation, 2020b. Common vulnerability and exposure. http://cve.mitre.org.
Mouratidis, H., Giorgini, P., 2007. Secure tropos: a security-oriented extension of the tropos methodology. Int. J. Softw. Eng. Knowl. Eng. 17 (02), 285–309.
Naver Corporation, 2016. NAVER Korean dictionary. http://krdic.naver.com/detail.nhn?docid=35726100.
Paja, E., Dalpiaz, F., Giorgini, P., 2015. Modelling and reasoning about security requirements in socio-technical systems. Data Knowl. Eng. 98, 123–143.
Pennock, M.J., Wade, J.P., 2015. The top 10 illusions of systems engineering: A research agenda. Proc. Comput. Sci. 44 (C), 147–154.
Ponemon, L., 2011. Cost of Data Breach Study: United States. Tech. Rep, Ponemon Institute, Traverse City, MI.
Prud'hommeaux E, S.A., 2008. SPARQL query language for RDF. (Accessed 16 March 2020).
Sindre, G., Opdahl, A.L., 2001. Capturing security requirements through misuse cases, nik 2001, norsk informatikkonferanse 2001. http://www.nik.no/2001.
Smith, E.A., 2006. Effects-based operations. Secur. Chall. 2 (1), 43–62.
Symantec Security Response, 2015. Butterfly: Corporate Spies Out for Financial Gain. Tech. Rep., Symantec.
Van Lamsweerde, A., 2004. Elaborating security requirements by construction of intentional anti-models. In: Proceedings. 26th International Conference on Software Engineering. IEEE, pp. 148–157.
Yu, E.S., 1997. Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering. IEEE, pp. 226–235.

Mr. Bong-Jae Kim graduated Korea Military Academy and received MS in the Dept. of Network Centric Warfare at Ajou University in Republic of Korea. His research interests include network, information security and software engineering.

Dr. Seok-Won Lee is a Full Professor and Chair of the Dept. of Software and Computer Engineering and Head of Graduate School of Software at Ajou University in Republic of Korea. He was a faculty member at the University of Texas at San Antonio and University of North Carolina at Charlotte in USA. He also worked at Science Applications International Corporation (SAIC) and IBM T.J. Watson Research Center as a senior research scientist. His areas of specialization include software engineering, knowledge acquisition, machine learning, and information assurance. He has published more than 180 peer reviewed articles. He is a member of IEEE, ACM and AAAI.