Analytical Study of Cognitive Layered Approach for Understanding Security Requirements using Problem Domain Ontology

Bong-Jae Kim* and Seok-Won Lee†
*Dept. of Network Centric Warfare, †Dept. of Software and Computer Engineering
Ajou University, Suwon City, Republic of Korea, 16499
*drzakal@ajou.ac.kr, †leesw@ajou.ac.kr

***Abstract*—Socio-technical Systems (STS) consist of complicated requirements that consider a variety of stakeholders' viewpoints, and are inherently complex due to heterogeneity characteristics of STS components. However, security in STS is still a major issue, which can be explained by the resulting cost and the impact of the STS intrusion on the whole enterprise. However, research related to recommending security requirements for a target STS is insufficient. Firstly, systematic acquisition of understanding the problem with rich context-awareness is not provided to STS, since the knowledge for the development and execution of STS is scattered. Secondly, the majority of security analysis focuses on only the technical approach, although it is necessary to perform a holistic analysis of STS due to heterogeneity characteristics. In order to solve these problems, we conduct a study of the three-layered framework for recommending security requirements through goal-oriented risk assessment using a Problem Domain Ontology (PDO). By using this framework, we demonstrate how the PDO is built through collecting, analyzing, and categorizing different information and knowledge from various sources, and how security requirements are recommended from the threat analysis and the goal-oriented risk assessment based on PDO. In addition, we discuss the applicability of this framework with a case study based on a real threat scenario. This paper contributes to security requirements engineering research by proposing a methodology for systematically organizing knowledge with a security requirements recommendation framework using the PDO.**

***Index Terms*—security requirements, problem domain ontology, socio-technical systems.**

I. Introduction

Socio-technical Systems (STS) consist of complicated requirements that consider a variety of stakeholders' viewpoints, and are inherently complex due to heterogeneity characteristics of STS components. The complexity of STS is derived from a complicated linking of various elements, such as business, systems, technologies, etc. Security issues, however, in STS are still especially significant problems, which can be explained by the resulting costs of STS intrusions and the resulting impact on the whole enterprise. The damages of intrusion have tremendous consequences in lost time, and increased costs and efforts to recover. [1]

In order to tackle this problem, it is essential to know why security issues in STS are emerging in the present day. Firstly, considerations in developing and executing STS are scattered. The complexity in STS comes from evolving threats, human aspects, business, laws, system services, system goals, etc. In addition, it is necessary to consider not only the number of stakeholders, but also the wide scope of stakeholder's fields. Even these issues add exponentially more complexity to STS. Furthermore, requirements from mere analysis of technological aspects are not satisfied by diversified stakeholders. Since the security quality attribute often leads to a negative effect on other quality attributes, such as performance, it is necessary to consider and negotiate with other quality attributes and various stakeholders. If these efforts are not well coordinated, it may lead to critical requirements defects. [2], [3]

Based on these understanding, we address following problems in this paper. Firstly, knowledge for design, development and execution of STS is scattered through various stakeholders. Even though there are plenty of related sources of knowledge available, it is very difficult to identify, model and analyze the relationships between pieces of information and create useful knowledge. Moreover, lack of such knowledge results in false analysis that eventually reveals significant requirements defects. Next, even though it is necessary to perform holistic analysis, due to the heterogeneity characteristics of STS, the majority of security analysis focuses on only the technical approach. Context-awareness for the heterogenic characteristics of STS can be derived from the consideration of multiple perspectives on business, human, application, and technology. When ignoring these perspectives, the analysis of security requirements for the given target STS will only focus on a single facet of problems, which makes harder to understand and estimate the holistic behavior from the emergent properties.

The aim of our research goal is to design and develop a framework to understand and recommend security requirements using the risk assessment from the threat analysis in STS with a knowledge-based Problem Domain Ontology (PDO). PDO [4] provides agreements among pieces of knowledge, creates explicit information, reuses and shares the common knowledge among stakeholders. The proposed three-layered framework consists of physical, information modeling and cognitive layers that encompass various meta-models for reasoning about the phenomena with the recommendation of an appropriate set of requirements. Based on this framework, we are able to better understand the security problems and recommend security requirements with rich contexts related to STS

1530-1362/16 $31.00 © 2016 IEEE
DOI 10.1109/APSEC.2016.37

derived from goal-oriented threat-analysis, risk assessments, and the related process under this framework.

In the following, Section II discusses the background and related work as the foundation of this research. Section III provides the introduction for the proposed framework. Section IV is the case study based on a real scenario. Finally, we conclude our paper and provide future developments in Section V.

II. Related Work

A. Security Requirements Modeling Methodology

The categorization of the security requirements modeling is well described in Fabien et al. [5] The multilateral approaches for security requirements engineering are proposed in [6] and [7]. Unified Modeling Language (UML) based modeling for security requirements is proposed in [8], [9], and [10]. Goal-based approach for security requirements is proposed by various researchers, including Secure Tropos [11], GBRAM [12], STS-Modeling Language [13], and the three-layered approach by Li et al. [14]. However, their modeling methodologies do not present systematic methods to organize and reuse knowledge for understanding and recommending security requirements. In addition, since the nature of elements that consist of security requirements models are different each other, it is necessary to have a framework that integrates various concepts for fully understanding the behavior of the models. Also, each modeling method does not provide an explicit specification method. In the case of using Common Criteria (CC) [15], it has an advantage in that it is reusable for the security requirements concepts. Even though Mellado et al. [16] proposed the Security Requirements Engineering Process (SREP) for CC, they did not demonstrate the organization of knowledge and modeling with the rich context of security requirements.

B. Risk Assessments

Special publication by National Institutes of Standard and Technology (NIST) [17] and Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [18], [19] are proposed for the risk assessments for the enterprise. Each of them proposed their own methodological framework using a template and process. These works were another foundations of this research by improving the utilization of evidences and artifacts with the security requirements engineering processes.

C. Problem Domain Ontology

The methodology for creating PDO in the security domain is proposed in [4]. Onto-ActRE [20] is a requirements engineering framework that integrates various modeling methods by using PDO ontology and use them to effectively construct security requirements for US Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP). Especially, their efforts are used for solving various problems related to the knowledge distribution, dispersion, diversification, and integrating such knowledge to build ontology.

Lastly, Kim et al. [21] proposed the conceptual framework using the cognitive three layers and the goal model in order to build security requirements ontology. This methodology showed the applicability of threat analysis and risk assessment using their ontology.

III. PIC Framework Based on PDO

A. Overview and Conceptual Model

This section provides the PIC framework based on PDO. This framework is developed with the goal model by providing holistic analysis based on various contexts from knowledge

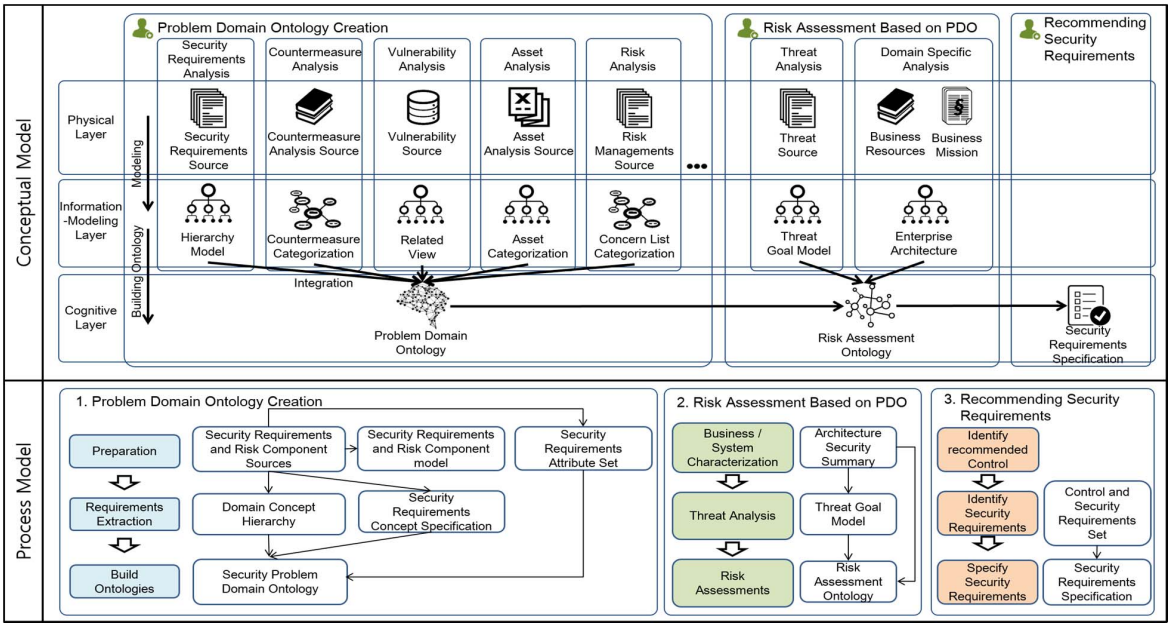


Figure 1. PIC Framework for security requirements based on Problem Domain Ontology

bases. Fig 1 shows the overview of the framework based on the Conceptual Model. We suggest three layers related to human's cognition: a **P**hysical layer, an **I**nformation-modeling layer, and a **C**ognitive layer. The physical layer represents phenomena or events from various sources in the real world. The information-modeling layer provides various types of models and their hierarchical representations using sources from the physical layer through modeling and analysis. Lastly, the cognitive layer provides the whole problem/situation context with a holistic understanding by using the relationships among concepts from the previous layers. The final goal of this framework is to understand and recommend security requirements with rich contexts by using PDO.

B. Process Model

The Process Model in Fig 1 demonstrates whole process of this framework. We divide the process into three phases: Problem Domain Ontology Creation, Risk Assessment Based on PDO, and Recommending Security Requirements. The PDO Creation Phase is used for creating the reusable ontology applied to whole security area. However, the phase 2 and 3 use more domain specific knowledge by identifying risks in related domains. This process can be customized for use depending on the domain.

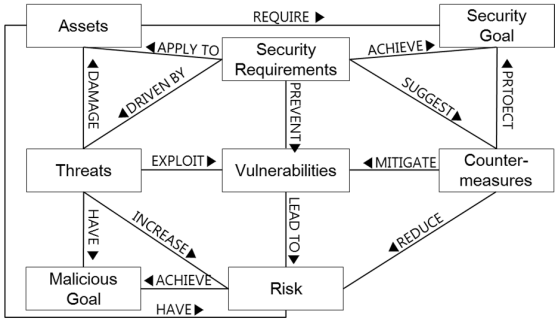


Figure 2. Security Requirements and Risk Components Relationship Model

1) Phase 1: Building Problem Domain Ontology

a) Preparation Step

Identifying Security Requirements Concepts and Sources: This sub-step identifies concepts for security requirements from various sources. Sources, such as documents or related knowledge, are used as the identification of concepts for security requirements. Concepts and their relationships for security requirements are applied and customized from the Security Requirements and Risk Components Relationship Model in [4], as shown in Fig 2. However, we notice that the *Goal* plays the key role in the threat case, because threats by malicious actors have their malicious goals. So, we added *Security* and *Malicious Goal* as symmetric concepts in this research.

In addition, it is necessary to collect related sources, such as documents in agency and institutes. In this research, we use ISMS C&A Process Guidance [22], ISMS Risk Management Guidance [23] by KISA, Common Weakness Enumeration (CWE) [24], Common Attack Pattern Enumeration and Classification (CAPEC) [25] by MITRE Corporation, and law information in Korea. [26]

Creating the Relationship between Concepts and Sources: This sub-step classifies and categorizes concepts with sources. Security Requirements are gathered from certification elements in the ISMS Certification and Accreditation (C&A) Process, and organized based on the same source. Countermeasures are mapped with the Control List in ISMS C&A Process, and Vulnerabilities are connected with CWE. The Risk comes from the Concern List in ISMS Risk Management Guidance, and the Asset is categorized using ISMS Risk Management. We map the threat with CAPEC. The visualization of this step is shown in Fig 3.

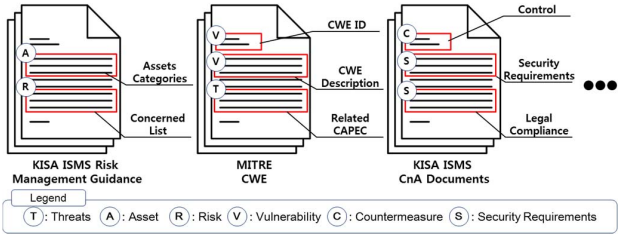


Figure 3. Identify concepts from related sources

Identifying Security Requirements Attributes and categorization: This step identifies security requirements attributes, which define more specific elements related to concepts for context-awareness through requirements analysis. Some attributes such as security goals and attack vectors are shared with other concepts. Table I shows security requirements attributes. These attributes are extendable based on sources and knowledge. We define security requirements using questionnaires in the C&A process.

TABLE I. Security Requirements 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

b) Requirements Extraction Step

This step is used for categorizing and creating hierarchies of all the elements in security requirements. Categorization and modeling for concepts are performed using Fig 3. Each of the marked elements is mapped to the related concept. For example, Security Requirements are derived from controls or countermeasures as the sub-category in the ISMS.

After building the hierarchy, all elements in this hierarchy are marked with an identification number in order to manage concepts and relationship in the PDO. In the cases of CAPEC or CWE, which have their own numbering, we follow their numbering methods. On the other hand, where is no numbering format, we give an ID number based on their concepts, attributes, and categorization. The asset is categorized into 10 items with an ID number. The countermeasure is marked by related asset and countermeasure type. For example, *Server Access Control* in access control is marked as SE-1 related to Server as an asset name, and PC-1 is related to Prevent Confidentiality

(PC) as a type of countermeasure. Based on each ID number, the entire set of elements in the hierarchy will be specified based on each source. In the case study, an example specification will be shown in further detail. After specification, the relationship between the elements is made based on Fig 2. With the understanding of attributes, specifications in each source are enriched for the Problem Domain Hierarchy. The result of making these relationships is shown in Fig 4.

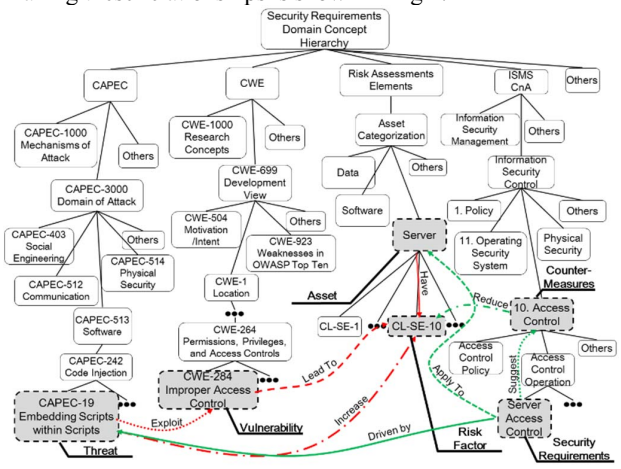


Figure 4. Security Requirements Concept Hierarchy and Relationship

c) Building Ontology

This step builds the PDO based on artifacts and models. Ontology elements are built with concepts and relationships using various tools, such as Protégé [27]. Fig 5 shows the PDO frame model with attributes. Attributes are divided into two types, Shared Attributes, and Dependent Attributes. A Shared Attribute is defined as the attribute that is shared with other concepts such as the Malicious Goal, Attack Vector, etc. A Dependent Attribute is only presented from the related concept, such as the Related Stakeholders and Description. Moreover, we also divide attributes into two categories: General Purpose Attribute, and Domain Specific Attribute. The entirety of these attributes contributes to the frame-model for the ontology used in the whole process.

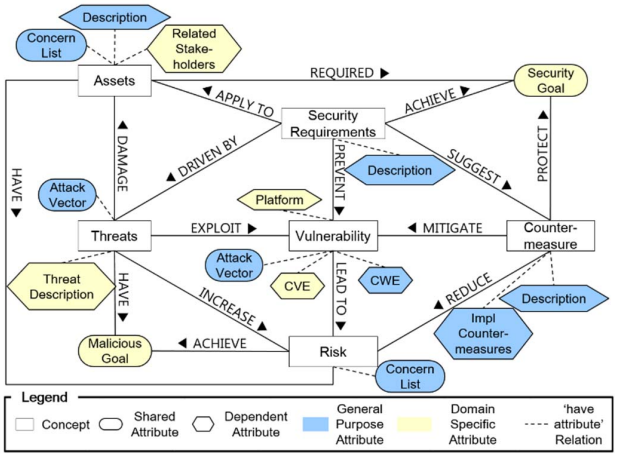


Figure 5. Problem Domain Ontology Frame Model

2) Phase 2: Risk Assessment based on PDO

This phase is related to the Risk Assessment in a target domain. Even though the basic process for risk assessment is similar to NIST Publication by Blank et al. [17], this framework focuses on the risk assessment by using PDO.

a) Business / System Characterization

Firstly, all information is collected from the architecture, which is related to security requirements from the business and system analysis in the chosen domain. It is assumed to have the Enterprise Architecture in the domain. By using views in certain architecture, this step identifies *Related Stakeholders, Asset, and Countermeasures* by summarizing security features.

b) Threat Analysis

Based on this information, the threat analysis is conducted with ontology proposed by Kim et al. [21], which analyzes the *Threat, Attack Vector, Asset, and Malicious Goal* using Goal-Based Modeling with the customized notations in Fig 6. This research uses i* framework proposed by Yu et al. [28], and several notations are customized for this framework.

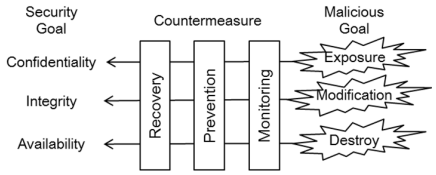
Figure 6. Customized Notations for threat analysis

c) Risk Assessments

We propose the Goal-Oriented Risk Assessment approach using the Security Attribute Triad. Fig 7 (a) shows that the triad consists of the security goals of *Confidentiality, Integrity, Availability*. We also define the malicious goal against the respective security goal. Firstly, *Exposure* damages *Confidentiality*, *Availability* is impacted by *Destroy*, and *Modification* hurts Integrity. These are the top levels of malicious goals that are achieved by the malicious actor. The risk assessment considers two elements for estimation: Potential Loss and Risk Probability. Risk Probability is considered *High* when the required Countermeasure related to a Security Attribute is not available, and *Low* when available. Potential Loss is considered when related threats achieve the malicious goal against related assets.

Based the concept model, Fig 7 (b) shows how to estimate the security goal for risk assessment. Firstly, the security attribute, malicious goal, and countermeasure type related to asset are set. Countermeasure type is categorized into three categories based on Fault Tolerance Mechanisms by Ricky Butler [29]: *Monitoring*, *Prevention*, and *Recovery*. Based on this categorization, the criticality of countermeasure type is decided using the system of *C* (Critical) and *N* (Not Required). *C* means the highest priority that related countermeasure type must be required, and *N* means lowest priority that is not required for specific type of countermeasure. As a result of this marking, each security goal is represented with *RP* (Required Prevention), *RM* (Required Monitoring), or *RR* (Required Recovery) based on the countermeasure type related to the asset. In the case of the criticality level *C*, it means all of the countermeas-

ure types are required. Fig 7 (b) demonstrates the way to calculate security goals for *Log Data*.



(a) Risk Assessment Concept Model

Security Attr.	Malicious Goal	Asset : Log Data			Security Goal
		Monitoring	Prevention	Recovery	
C	E	N	C	N	RP
I	M	C	C	C	C
A	D	C	N	C	RMRR

(b) Risk Assessment Goal Calculating Matrix

Figure 7. Risk Assessment Concept Model and Goal Calculating Matrix

Fig 8 (a) shows the risk assessment process using goal calculation. The first step is to decide on the status of the *Risk Exist* based on the system platform or configuration. By using platform information with related version information, the possibility of threat is identified. In addition, weak security configurations, such as a weak password, and weak encryption mechanisms, are considered to decide the probability level of a threat. If probability of threat is identified as *Risk Exist*, risk assessment is performed using the template in Fig 8 (b). This template gives guidance to decide the specific risk based on the security goal, and provides the required countermeasure type.

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 Intent			Current Countermeasures										Risk Factor			Recommen-dations
	C	I	A			E	M	D	C			I			A				E	M	D	
									M	P	R	M	P	R	M	P	R					

(b) Risk Assessment Template

Figure 8. Goal-Oriented Risk Assessment Process and Template

Step 3 defines the Asset information with a Security Goal from the Goal Calculating Matrix. Then, Step 4 represents the result of the Threat Analysis using a Goal Model with threat, vulnerability, etc. A malicious goal is defined as *E* (Exposure), *M* (Modification), and *D* (Destroy). *E* means that the related asset is threatened with *Exposure to Malicious Actor* by a threat without appropriate countermeasures. Step 5 maps the current countermeasures to *Monitoring*, *Protection*, and *Recovery* based on architecture or Security Service Summary. These are marked as *Yes* or *No*, but if it is not normally operating or the real security service is not the same as the required countermeasure, it is also marked as *Weak* (W). Then, the risk factor is decided in Step 6. The required countermeasure in the asset security goal is compared with current countermeasure. For example, when the availability of the asset is marked as *C*, all of the related countermeasures are satisfied. Moreover, when *Recovery of Availability* in current countermeasure is estimated as *W*, *Destroy* in the risk factor is marked as *Y*. Through this estimation, we identify the risk factor by asset, threat, and current countermeasures. Lastly, Step 7 concludes recommended countermeasure types based on the Security Goal.

3) Phase 3: Recommending Security Requirements

Having the identified risks from the previous phase, Recommending Security Requirements is performed in Phase 3. Identified risks derive the recommended countermeasure which can elicit the security requirements. Through the inference using ontology and templates, it leads to security requirements specifications with attributes, and provides the whole context related to the security requirements.

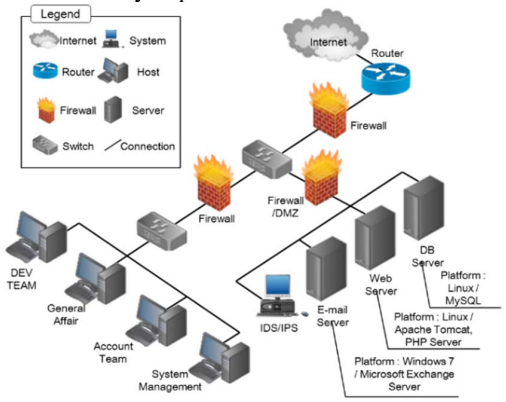


Figure 9. Network Topology for Case Study

IV. Case Study

A. Case Study Scenario

1) Case Study Scenario: Butterfly (Hacktool.Eventlog)

We use a real threat scenario for the case study, Hack tool.Eventlog. This is a threat tool used for intrusion against technology or pharmaceutical companies with financial purposes by Butterfly since 2012. Main targets for intrusion are the email server, or the enterprise content management server. Hacktool.Eventlog is specifically a tool for parsing, dumping, and deleting log data including self-destruction. The case study is to show the process of building PDO, Risk Assessment, and Recommending Security Requirements specifically related to Hacktool.Eventlog. We use a Security Response Report by Symantec [30] for analyzing threat, and assume that this is a Zero-Day Vulnerability even though it has already been solved.

2) Case Study Environment Introduction

We will also build a virtual environment for this framework. First, the name of company in this scenario is BK Company, which develops mobile software. As the network topology for

this company is shown in Fig 9, this company owns Servers in the DMZ Network, and several teams to develop and manage. The current countermeasures that are operated in the companies are partially summarized in Table II.

TABLE II. CURRENT SECURITY ENVIRONMENTS SUMMARY

Asset		Type	S. A.	Related Countermeasure	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	Log Backup	• Back up the Log data into external media
	Database 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
⋮	⋮	…	…	…	…

B. Framework Application

1) Building Problem Domain Ontology

This sub-section shows building the ontology using the PDO process introduced in section 3. The preparation step is skipped due to the fact that it would contain the same contents as the previous section. In the case of the Requirements Extraction Step, we focus on the File Integrity Monitoring (FIM) as a countermeasure against Hacktool.Eventlog. FIM is the process or control system that monitors the integrity of the file. FIM is marked the ID number, C-DA-1-MI-1, as a member of Monitoring Integrity (MI) for Log Data (DA-1). Implementable solution for this countermeasure is added. This countermeasure is mapped to the Log and Backup system in ISMS Certification for security requirements, SR-11-6-2. Related Assets such as Server and Data makes an *Apply to* relationship with this requirement. Even though the weaknesses that are mitigated by this countermeasure are varied, we specify Expected Behavior Violation for this case study, CWE-440. CAPEC elements that exploit this Weakness are the Audit Log Manipulation, CAPEC-268. Lastly, this countermeasure reduces Risk, Mistreat Data, CL-DA-7. Fig 10 (b) represents the specification of these elements. After the Requirements Extraction Step, recommendations are created, like Fig 5. Lastly, Fig 10 (a) shows the partial ontology for this case study.

2) Risk Assessment based on PDO

We conduct the risk assessment using the threat scenario based on the PDO. By using the correlation between the goal model and the ontology by Kim et al [9], Security Goal-Based Model (SGBM) which used i* Framework in Fig 11 (a) is created. Concepts in SGBM are mapped to Risk Assessment Ontology including Related Stakeholders and Security Goals from Security Architecture. The result of the ontology is represented in Fig 11 (b).

By using the entire information, two steps for estimating risk existence are performed. Through comparing with platform information and security configuration, risk existence in BK Company is identified. In this case study, we assume that the platform information is the same as the required platform information of the threat. After identifying the *Risk Exist*, risk assessment using the template is performed, Fig 11 (c). Since the case study threat affects the *Log Data*, the security goal is marked with the required countermeasure type by use of a Goal Calculating Matrix in Fig 7 (b), such as a *C* in Integrity. Furthermore, the threat, attack vector, vulnerability, and malicious intent are identified from the ontology as Step 4. Then, current countermeasures are identified using a Summary of Countermeasure in Table II. Step 6 estimates the Risk Factor using current countermeasures, Security Goal, and Malicious Intent. For example, Security Goal of Integrity in Fig 11 (c) requires all related countermeasures. Current countermeasures for Integrity are marked as N, which means that no matching countermeasure exists. Then, the risk factor is determined whether it is Modifiable and/or Destroyable based on the malicious goal. Based on the risk factors, the recommended countermeasure Type is concluded based on Security Goal, which is not currently operating in BK Company.

3) Recommending Security Requirements

After completing the Risk Assessment, security requirements are specified using the recommended countermeasure type and related concepts with the template and ontology. Fig 11 (d) demonstrates the Security Requirements Specification. Firstly, the Log Data in Asset has countermeasures categorized into Security Attributes. Each attribute has their implementable countermeasure as solutions, such as FIM, OS Patch, etc. After choosing FIM as a countermeasure for Monitoring Integrity, Security Requirements are specified in the template. Security requirements with contexts of various elements are specified within the template, including Recommendations in Fig 11 (d).

C. Discussion

We demonstrate the case study by using the proposed framework with a literature approach for the applicability of PDO. Based on this research, we can extend the framework with prototype development in order to further contribute to security requirements engineering research by proposing a methodology for systematically organizing knowledge with a security requirements framework using a PDO.

However, we also identify our limitations in this research. Firstly, ontology with various areas of knowledge has a scalability problem due to increasing the number of concepts and relationships, which also leads to the maintenance problem of the ontology. In order to maintain this ontology, expert-level SME knowledge is necessarily required. In addition, it leads to potential requirements defects caused by incorrect knowledge management due to the human error during building of ontology. Even though the risk is assessed using Goal-Oriented Assessments, it is still challenging for risk assessment with a more complicated context. Lastly, experimental evaluation needs to be conducted to consolidate the framework.

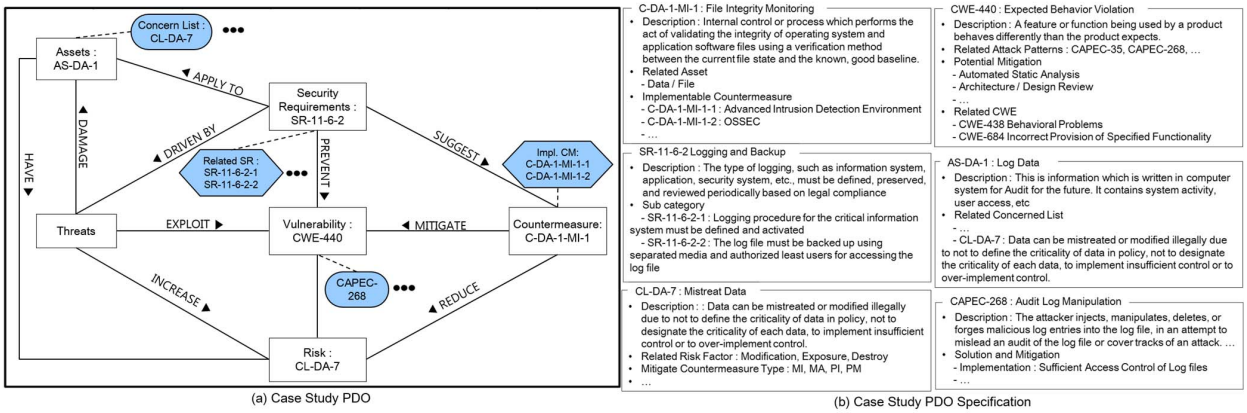
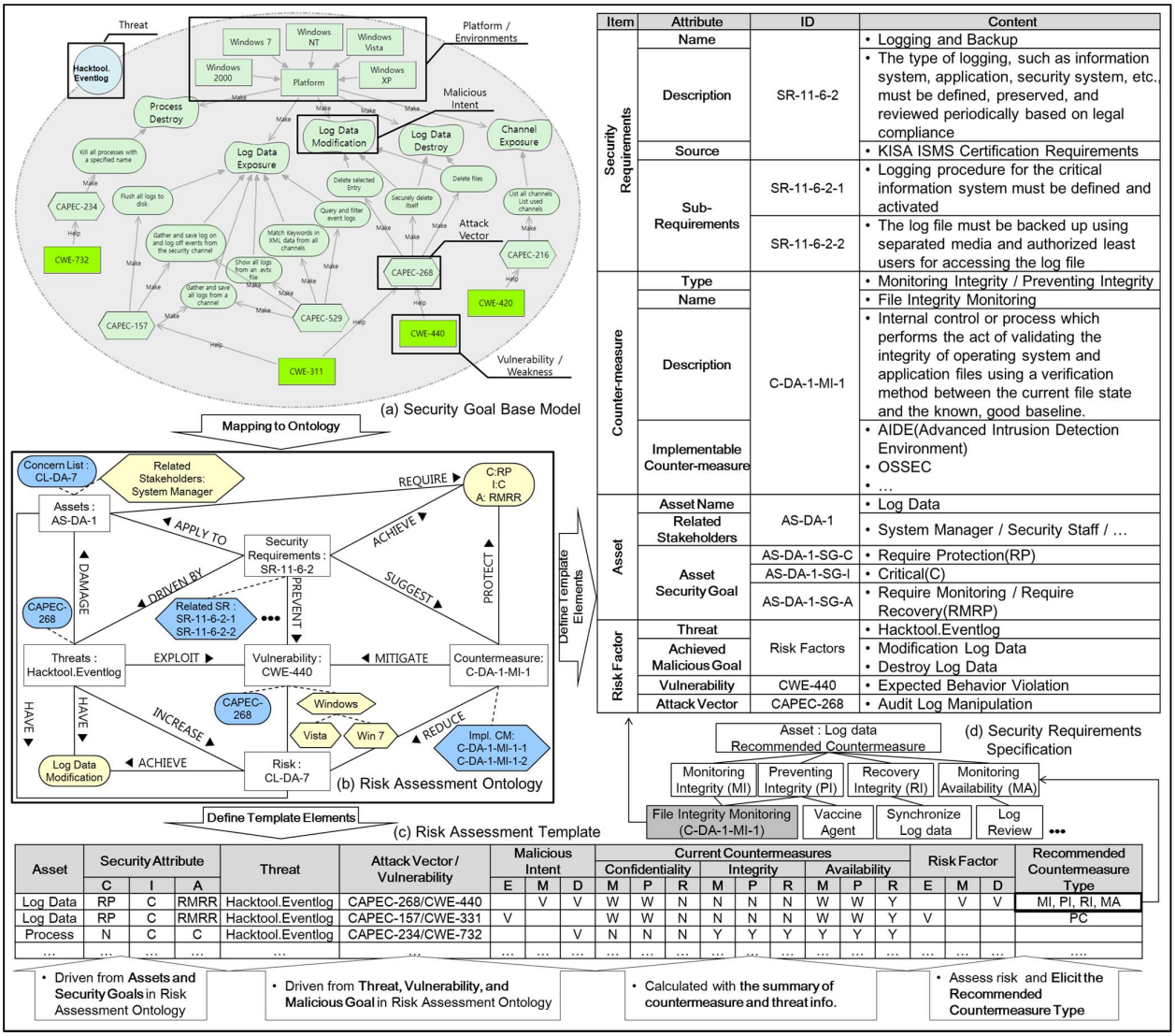


Figure 10. Case Study Problem Domain Ontology and Specification



Item	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-MI-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	• 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	Risk Factors	• Hacktool.Eventlog
	Achieved Malicious Goal		• Modification Log Data • Destroy Log Data
	Vulnerability	CWE-440	• Expected Behavior Violation
	Attack Vector	CAPEC-268	• Audit Log Manipulation

Asset	Security Attribute			Threat	Attack Vector / Vulnerability	Malicious Intent			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				
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

- Driven from **Assets and Security Goals** in Risk Assessment Ontology
- Driven from **Threat, Vulnerability, and Malicious Goal** in Risk Assessment Ontology
- Calculated with **the summary of countermeasure and threat info.**
- Assess risk and **Elicit the Recommended Countermeasure Type**

Figure 11. Framework Case Study based on Problem Domain Ontology

V. Conclusion and Future Work

We have demonstrated the proposed framework by building a PDO, conducting Risk Assessment, and Recommending Security Requirements. In addition, we conducted a case study with a real threat scenario in order to show the applicability of this framework. This paper especially contributes to strengthening the understanding of security requirements using a well-organized knowledge base, and to analyze with holistic approach in security requirements engineering perspective. In the near future, we will propose a prototype for this framework with an empirical study in order to solidify this framework.

Acknowledgment

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2013M3C4A7056233).

Reference

[1] I. Sommerville, "Sociotechnical systems," in *Software Engineering*, Pearson, 2009, pp. 263-288.

[2] P. D. Collopy, "A Research Agenda for the Coming Renaissance in Systems Engineering," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Nashville, 2012.

[3] M. J. Pennock and J. P. Wade, "The Top 10 Illusions of Systems Engineering: A Research Agenda," in *2015 Conference on Systems Engineering Research*, 2015.

[4] S.-W. Lee, R. Gandhi, D. Muthuranjan, D. Yavagal and A. Gail-Joon, "Building Problem Domain Ontology from Security Requirements in Regulatory Documents," in *Workshop on Software Engineering for Secure Systems*, New York, 2006.

[5] B. Fabian, S. Gurses, M. Heisel, T. Santen and H. Schmidt, "A Comparison of Security Requirements Engineering Method," *Requirements Eng,* pp. 7-40, 2010.

[6] S. F. Gurses, B. Berendt and T. Santen, "Multilateral Security Requirements Analysis for Preserving Privacy in Ubiquitous Environments," in *Workshop on Ubiquitous Knowledge discovery for users(UKDU 06)*, 2006.

[7] N. R. Mead and T. Stehney, "Security Quality Requirements Engineering(SQUARE) Methodology," in *Software Engineering for Secure Systems*, St. Louis, 2006.

[8] G. Sindre and A. L. Opdahl, "Capturing Security Requirements through Misuse Cases," in *The 14th Norwegian Informatics conference*, 2001.

[9] S. H. Houmb, S. Islam, E. Knauss, J. Jurjens and K. Schneider, Eliciting security requirements and tracing them to design: An integration of common criteria, heuristics and UMLsec, *Requirements Engineering 15(1)*, (2010) pp.63-93.

[10] T. Lodderstedt, D. Basin and J. Doser, "SecureUML: A UML-Based Modeling Language for Model-Driven Security," in *The 5th International Conference on the Unified Modeling Language*, London, 2002.

[11] H. Mouratidis and P. Giorgini, "Secure TROPOS: A Security-Oriented Extension of the Tropos Methodology," *International Journal of Software Engineering and Knowledge Engineering,* vol. 17, no. 2, pp. 285-309, 2007.

[12] A. I. Anton and J. B. Earp, "Strategies for Developing Policies and Requirements for Secure Electronic Commerce Systems," in *i* Workshop on Security and Privacy in E-Commerce at CCS2000*, 2000.

[13] E. Paja, F. Dalpiaz and P. Giorgini, "Modelling and Reasoning about Security Requirements in socio-technical systems," *Data and Knowledge Engineering,* pp. 123-143, 2015.

[14] T. Li and J. Horkoff, "Dealing with Security Requirements for Socio-Technical Systems: A Holistic Approach," in *Advanced Information Systems Engineering, CAiSE 14*, Thessaloniki, 2014.

[15] Common Criteria, "Part 1: Introduction and general model," in *Common Criteria for Information Technology Security Evaluation*, Common Criteria, 2012, pp. 38-44.

[16] D. Mellado, E. Fernandez-Medina and M. Piattini, "Applying a Security Requirements Engineering Process," in *ESORICS*, 2006.

[17] R. M. Blank and P. D. Gallagher, "Guide for Conducting Risk Assessments," National Institute of Standards and Technology, Gaithersburg, 2012.

[18] C. Alberts, A. Dorofee, J. Stevens and C. Woody, "Introduction to the OCTAVE Approach," Carnegie Mellon Software Engineering Institute, Pittsburgh, 2003.

[19] R. A. Caralli, J. F. Stevens, L. R. Young and W. R. Wilson, "Introducing OCTAVE Allegro: Improving the Information Security Assessment Process," Carneige Mellon Software Engineering Institute, Pittburgh, 2007.

[20] S. Lee and R. A. Gandhi, "Ontology-based Active Requirements Engineering Framework," in *Asia-Pacific Software Engineering Conference*, 2005.

[21] B. Kim and S. Lee, "Conceptual framework for understanding security requirements: A Preliminary study on Stuxnet," in *Asia-Pacific Requirements Engineering Symposium*, Wohan, 2015.

[22] Korea Internet and Security Agency, "Information Security Management System Certification Policy Guidance," Korea Internet and Security Agency, Seoul, 2013.

[23] Korea Information Security Agency, "Guide to information security management system risk management," Korea Information Security Agency, Seoul, 2004.

[24] The MITRE Corporation, "Common Weakness Enumeration(CWE)," Dec 2015. [Online]. Available: http://cwe.mitre.org. [Accessed 10 June 2016].

[25] The MITRE Corporation, "Common Attack Pattern Enumeration and Classification(CAPEC)," Nov 2015. [Online]. Available: http://capec.mitre.org. [Accessed 10 June 2016].

[26] Ministry of Government Legislation, "National Law Information Center," 2015. [Online]. Available: http://www.law.go.kr/eng/engMain.do. [Accessed 10 June 2016].

[27] M. Horridge, "A Parctical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.3," The University of Manchester, Manchester, 2011.

[28] E. S. K. Yu, "Toward Modelling and Reasoning Support for Early-Phase Requirements Engineering," in *International Symposium on Requirements Engineering*, Annapolis, 1997.

[29] R. W. Butler, "A Primer on Architectural Level Fault Tolerance," NASA Center for AeroSpace Information, Hampton, 2008.

[30] Symantec, "Butterfly: Corporate spies out for financial gain," Security Response, Mountain View, 2015.