Conceptual Framework for Understanding Security Requirements: A Preliminary Study on Stuxnet

Bong-Jae Kim[1(✉)] and Seok-Won Lee[2]

[1] Department of Network Centric Warfare, Ajou University, Suwon, Republic of Korea
drzakal@ajou.ac.kr
[2] Department of Software Convergence Technology,
Ajou University, Suwon, Republic of Korea
leesw@ajou.ac.kr

Abstract. As we analyze the latest occurring Advanced Persistent Threats (APT), we understand that attackers tend to conduct their methods by various means and ways, and are not limited to just a single pattern. Even though threats are represented by using a variety of modeling techniques, very little research has been done to learn about the security requirements needed from the understanding of commonly rooted causes of these threats. We propose a layered conceptual framework to better understand the problem from specific instances to generalized abstractions, so that it can eventually provide a good set of security requirements. In this framework, we propose building the ontology based on the Goal-based Model and Activity Diagram, and showing how to understand the context of the problem domain with extended relationships between concepts. We expect that our work can provide a foundation of good insight of the problem domain of security requirements and that we elicit security requirements through this work based on a preliminary case study on Stuxnet.

Keywords: Security concept requirements framework · Security ontology · Security Goal-based model · Attack activity diagram

1 Introduction

Recently, many response reports that analyze a variety of threats and attacks are provided by information security companies and government. Unlike past years, recent attack patterns in reports involve various elements such as networks, software, physical elements and human activity to achieve its malicious goal of an Advanced Persistent Threat (APT). In addition, these threats exploit one or more Zero-Day Vulnerabilities as parts of its attack. [1] Since it is difficult to predict the means and ways of such threats, it is essential to develop a way to understand the nature of the threats with possible means to overcome the said threat.

Although various threat modeling approaches are proposed [2], there are some limitations to creating requirements from these models. The majority of modeling techniques just concentrate on technical analysis, while research about extracting, eliciting, and inferring common and generalized requirements is insufficient. Though

L. Liu and M. Aoyama (Eds.): APRES 2015, CCIS 558, pp. 135–146, 2015.
DOI: 10.1007/978-3-662-48634-4_10

there is much work to provide techniques for attack modeling, for example, identifying various attack routes with visualization and calculating the most plausible attack route by using the stochastic method in each route, the conceptual framework to elicit common concepts has not been yet identified in this area. [3]

Thus, it is essential to research and study framework that can create requirements based on common and generalized concepts, because it still needs to progress under a complex security environment. In this paper, we have conducted research on understanding the problem domain using defined relationships between concepts and extended relationships based on the security conceptual framework, and creating security requirements. Using this framework, we expect to understand the problem domain of security requirements, generate models using reusable concepts, and help to draw the countermeasure using a requirements engineering process.

We organize four more sections to discuss our conceptual framework. Section 2 provides the introduction of related works for our research. Section 3 introduces the conceptual framework for our research, and Section 4 outlines the case study of our work using Stuxnet. Lastly, Section 5 concludes our research and provides the application to future work.

2 Related Work

We have conducted literature surveys to learn about relationships between security requirements and threats from the past researches.

First, a goal-model for the security area is suggested by Elahi et al [4] using i* framework, and they provide the comparison between i* framework and other conceptual framework in order to show the adaptability of i* framework to generate the security requirements as a goal model. In addition, they extended the i* framework to represent security notations such as vulnerability, malicious goals, etc., and adapted it into the case study, the Guardian Angel, to validate the goal model adaptability in the security area.

Lin et al. [5] used the modeling method of privacy and security based on the Guardian Angel case study, and suggested the concept of extraction based on the scenario, using a dependency analysis model for making connections between attack analysis models and countermeasure analysis models.

Research on the ontological engineering have been actively progressive and there are several efforts related to the security. First of all, Li et al [6] proposed the ontology based on the condition changes and behavior to design a system that provides the alert against intrusive behavior. However, it is hard to figure out the concept from the example, and to generate the corresponding security requirements from this model.

Wang et al. [7] proposed an ontology to manage the vulnerability related to the Common Vulnerability Enumeration (CVE) and Common Platform Enumeration (CPE), which give us some useful elements of security ontology. However, they did not make connections with real threats. Their works provide us the inspiration of a research base as we try to adopt the real threats.

Kotenko et al. [8] proposed the implementation of ontology in the SIEM system in order to overcome the limitations of the relational database, and their work give to us an understanding of the advantages of the implementation of ontology.

Elahi et al. [3] represented the conceptual framework with ontology as the meta-data model. They compared multiple conceptual frameworks including the i* framework, which can model malicious behavior and vulnerabilities with ontology. They considered malicious actions and countermeasures in their ontology. However, the context awareness of the various variant of tasks needed to be considered.

Lee et al. [9] proposed the Problem Domain Ontology Process to identify the security requirements for DITSCAP Automation, the Onto-ActRE and the modeling process using four steps of the PDO process. Their works defined the metrics and measures, and generated the security requirements. Through their works on the relationship between process, modeling and ontology, we understand that the conjunction between concepts in models and ontology provides the ability to generate the security requirements based on the surrounding knowledge.

Finally, we select the Stuxnet case as a case study of our approach which is a good example of Advanced Persistence Threats from [1]. The study of the Stuxnet is referred on [10, 11]. They analyze the malware with a technical approach and we referred to [12] for the modeling of the case.

3 Proposed Conceptual Framework

This section will provide imitations of the previous research efforts and the introduction to the proposed conceptual framework to overcome the limitations.

3.1 Approach

The generation of security requirements in information security is very challenging due to the complex and diverse attack techniques. To specify the requirements in the requirements engineering process, it is necessary to understand the problem domain.

The proposed framework consists of three layers: a physical layer, information layer, and cognitive layer. First, the physical layer represents real-world events, accidents, phenomena, and results of attacks, etc. The information layer is the area that consists of the analysis and representation of physical layer through the components of assets, attacks, countermeasures, goals, etc. Lastly, the cognitive layer is an understanding of both the physical and information layer. This layer can provide reasoning on the business, political, economic, society, standard, policies, and regulations impacts of threats and events in the physical and information layers.

In order to understand the security problem domain, we will propose the ontology based on the above three layers which can perform the bridge of understanding between each model and security requirement, and can create relationships between extracted concepts, like in Fig. 1. As you can see relations between layers in Fig. 1, artifacts in the information layer is generated through modeling and analyzing threats in the physical layer, and elements of artifacts in the information layer are mapped to generate the ontology based on the meta-data model. This ontology can provide the context-awareness specification using relations between concepts, and generate the scenario model. The scenario model will be the foundation for eliciting possible threats or real scenarios.

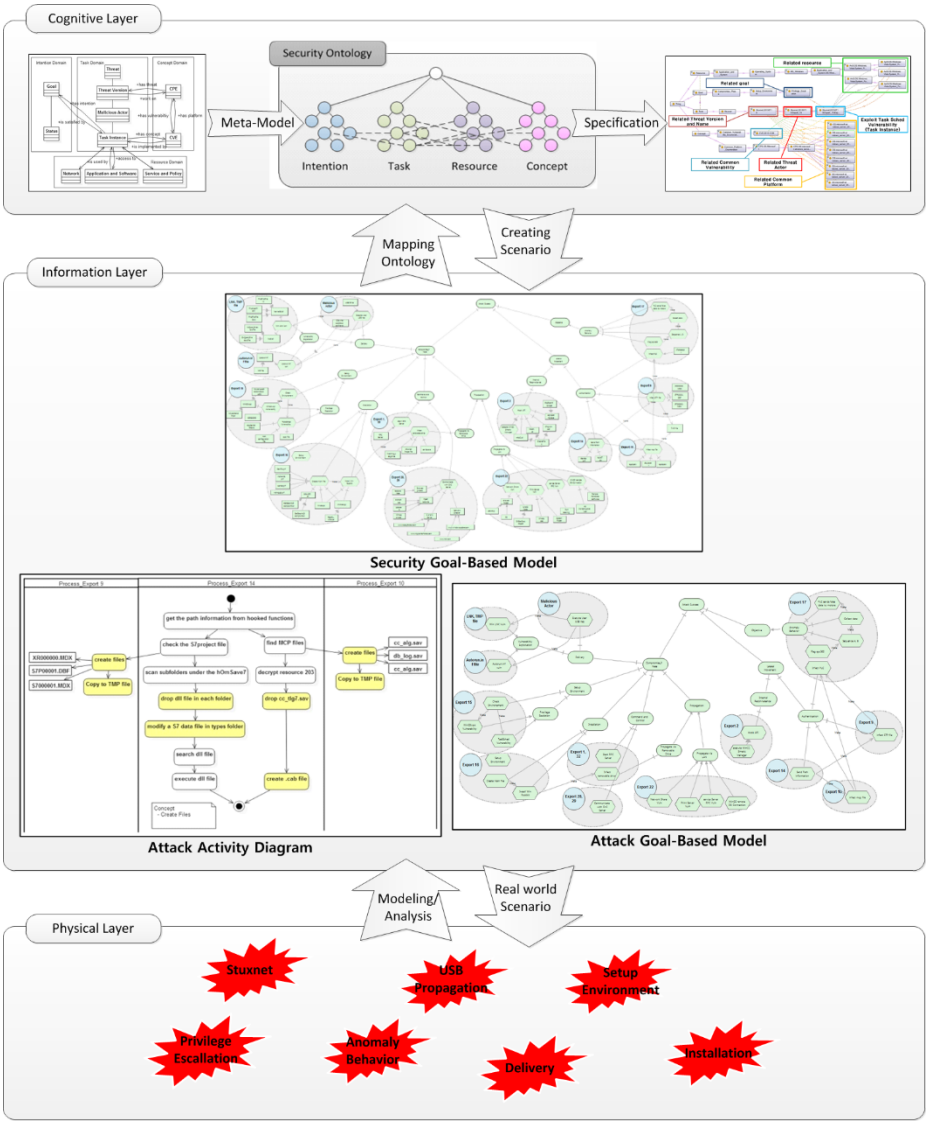

Fig. 1. The relationship between modeling and ontology based on layers

3.2 Overview of Security Concept Framework

In this sub-section, we briefly introduce our conceptual framework. As shown in Fig. 2, by comparing the Requirements Engineering Process with Stakeholder Requirements Definition Process in the System Security Engineering Process, published in the Special Publication 800-160 by the National Institute of Standard and Technology (NIST) [13], we will show how the framework can be implemented.

The Requirements Engineering Process consists of the following phases: Elicitation, Understanding, and Structuring, Modeling and Analysis, Communication and Negotiation, and Verification and Validation. This process is similar to the Stakeholder Requirements Definition Process in the System Security Engineering Process: Elicit Stakeholder Security Requirements, Define Stakeholder Security Requirements, and Analyze and Maintain Security Requirements. The conceptual framework performs part of this process and creates artifacts to understand the requirements during process. Ultimately, the purpose of the conceptual framework is to understand the related context of the given physical security situation, and to analyze the related security requirements components by considering the consequential results on laws, regulations. Then, one can understand the overall risks through the integrated ontology.

First, underlying this process, the fundamental research to identify instances related to concepts must be conducted. Next, based on these instances goals, tasks, and actors are extracted with an Attack Goal-Based Model (AGBM) using the i* framework, one of goal-based models in the requirements engineering. Then, an Attack Activity Diagram (AAD) is created, which can extract activities, related vulnerabilities, and resources based on identified task from the AGBM. Through mapping between elements in AGBM and AAD, a Security Goal-Based Model (SGBM) is generated, and instances related to security concepts in the SGBM are elements of the Security Ontology.

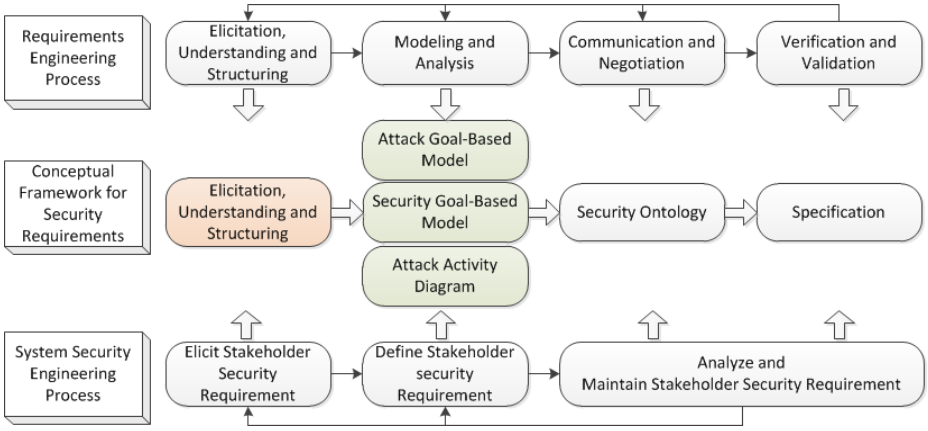

Fig. 2. Relationship with Requirements Engineering Process, System Security Engineering Process and Conceptual Framework for Security Requirements

Meta-data Model for Conceptual Framework

The Conceptual Framework is conducted based on the Meta-data Model in Fig. 3. These Security Concepts are elements in AGBM and ADD. Using this meta-model, we create the Goal-Based Security Model and finally the Security Ontology with the relationship among the domains.

First, we divide the domains into four categories: the Intention, Task, Concept, and Resource Domains. The Intention Domain includes both the Status and Goal knowledge with the hierarchical structure between them in the domain. The Goal is similar to the attack phase. We apply Bryant's work [14], which constructs the taxonomy of

the attack intention. The Task Domain includes the Threat, Threat Version, Actor, and Task Instance with a vertical hierarchical structure. Threat means the name of the threat in Task Domain, and it has the version as the subclass. Moreover, the Malicious Actor is something that performs the task to achieve one or more malicious goals within each version. In addition, Task Instance means the real task conducted by the Malicious Actor. Concept Domain includes the conceptual knowledge related to the threat, such as the vulnerability and the related platform that can be implemented for the attack. In our paper, we use the Common Vulnerability Enumeration (CVE) and Common Platform Enumeration (CPE). The Resource Domain is related to the means or resources used to conduct the task: the Service and Policy Layer, Application and Software Layer, and Network Layer. The Service and Policy Layer mean the policy or activity performed by people or organizations. The Application and Software Layer is the same concept as the application layer in TCP/IP Protocol Stack, such as software, process, OS, etc. Lastly, the Network Layer is the abstracted layer from transport to the physical layer in the TCP/IP Protocol Stack.

These Security Concepts makes relationships with each other. The solid line in the diagram means the vertically hierarchical connection. The arrow line indicates the relationship between different domains or different areas. These defined relationships are basis of the inference and assumption.

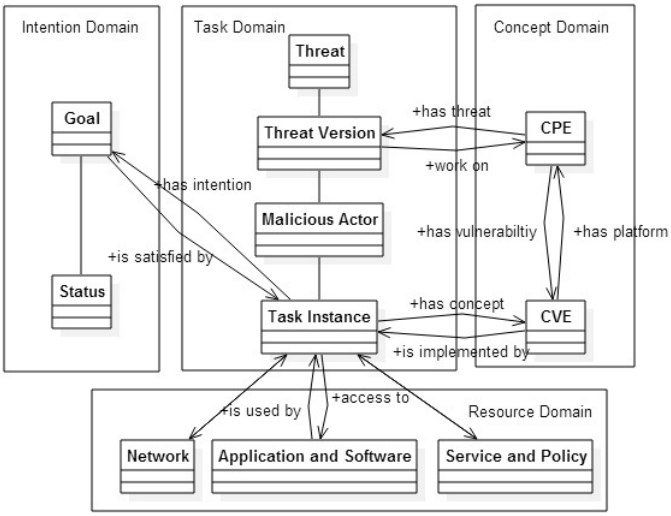

Fig. 3. Security Concept Requirement Framework Meat-Data Model

Each concept in this meta-data model came from elements of models in the information layer. Through eliciting elements during the framework, each element forms the ontology with relations between concepts in order to make aware the contexts of threats. For example, elements in the Intention and Task Domains came from AGBM which used the i* framework, and elements in Concept and Resource Domain came from AAD.

3.3 Attack Goal-Based Modeling

Previous works for attack modeling techniques were proposed in various forms, such as Attack Graph, Attack Tree, Bayesian Graph with a stochastic approach, and the Boolean Drive Markov Process (BDMP) models [2]. In the case of these models, they provided the modeling technique for specific events. However, as they focus on modeling with a technical analysis, there are limitations in the perspective of Requirements Engineering, such as limited representation of the intention and the trade-off value and reinterpretation for implementing the Requirements Engineering Process. To remedy these drawbacks, we propose the Attack Goal-Based modeling (AGBM) rooted from Goal-based modeling in the Requirements Engineering Process, which represent the intention and trade-off value.

Reasons why we choose the Goal-based model out of various requirements engineering models include the fact that people who have malicious intentions conduct almost all threats in cyber space, and these intentions are the foundation of tasks of customized malicious actor or techniques. To achieve the malicious goal, malicious people build certain methods and tools that can access resources to perform tasks and a series of activities. Though these elements of attack are implemented into codes and programs, we can extract concepts through a series of abstracting, modeling, and analysis of various attack scenarios to generate security requirements not only for these scenarios, but also for those that can be prevented early because generated requirements consider the common characteristics of various attack/threats scenarios.

Attack Goal-Based Modeling (AGBM) uses i* Framework with the OpenOME, the open source tool for goal-based modeling. Elahi et al. [4] conducted the research on frameworks comparison for the security requirement representation with other frameworks. They showed an explicit representation of relationships with goals, tasks, soft-goals, and resources and the extensibility for the security notation. Therefore, we choose i* framework because of strong points when illustrating the chained-attack using the embedded and extended notation. In this model, tasks will be related to the Attack Activity Diagram, and resources and related vulnerabilities will be shown.

3.4 Attack Activity Diagram

An Attack Activity Diagram (AAD) is the modeling process used to extract resources and concepts related to each task with serialized order of activities, and identify the anomaly behavior and interaction with resources. There are three reasons why we choose the Activity Diagram in UML for modeling the detail part. First, because we notice that the chained attack is conducted across multiple layers and has certain steps to perform the task, it is necessary to illustrate temporal-ordered events and behaviors that are difficult to represent in a goal model effectively. In addition, it needs to represent the multi-layered concept for understanding layered resource access. Lastly, the activity diagram gives the specification of the behavior process.

3.5 Security Goal Based Model

Security Goal-Based Model (SGBM) is the result of the integration with mapping instances in the AAD into the AGBM in order to understand the threat flow. Like the AGBM, it also uses the i* Framework and extends the previous work with embedded notations, resources and concepts from the AAD. Activities in AAD are represented as the concept of context-awareness with the interaction of resource. Therefore, the SGBM looks like the extended version of AGBM with adding resources and concepts. Because SGBM represents the specific event, this model needs to be converted into the ontology. In other words, SGBM constructs a bridge between Security Ontology, real events, and foundation of the ontology.

3.6 Security Ontology

This is integrated ontology based on the meta-data model extracting instances related to security concepts from previous models. Through using the ontology, we can define the relationship between security concepts. Advantages of the security ontology are that it provides a more effective comparison with the relational database model, in which it is difficult to change the scheme, gives the context awareness based on accessing resources and relationships with other information, and provides the inferences to understand other concepts from the specific concept. In addition, the specification of the problem domain can be means for the communication among related stakeholders, and be helpful in forming the common understanding for generating requirements. Protégé 4.3[15] based on the Meta-Data Model is the editor application for this ontology.

4 Case Study

This section provides the case study of the proposed conceptual framework, which shows the process from modeling to creating the ontology using the Stuxnet case in 2010. First, Stuxnet has the purpose of sabotaging the Iranian Nuclear Program in the Natanz Uranium Enrichment Plant, by exploiting the vulnerability of the Industrial Control Systems (ICS), Programmable Logic Controller (PLC), and Siemens Step-7 software with Windows systems that operate in the isolated network. They deliver the malware from an external environment using the thumb drive to plug into the isolated network, automatically propagate, and conduct the attack against inside and outer network. It also spreads worldwide with high infection probability via the Internet, more specifically located in the Middle East. It used several techniques, such as Zero-day vulnerability Exploitation, Windows Rootkit, PLC Rootkit, Antivirus Evasion Technique, Network infection routines, Command and Control Interface, etc. [11, 12] In this case study, we will briefly show the case of Export 15.

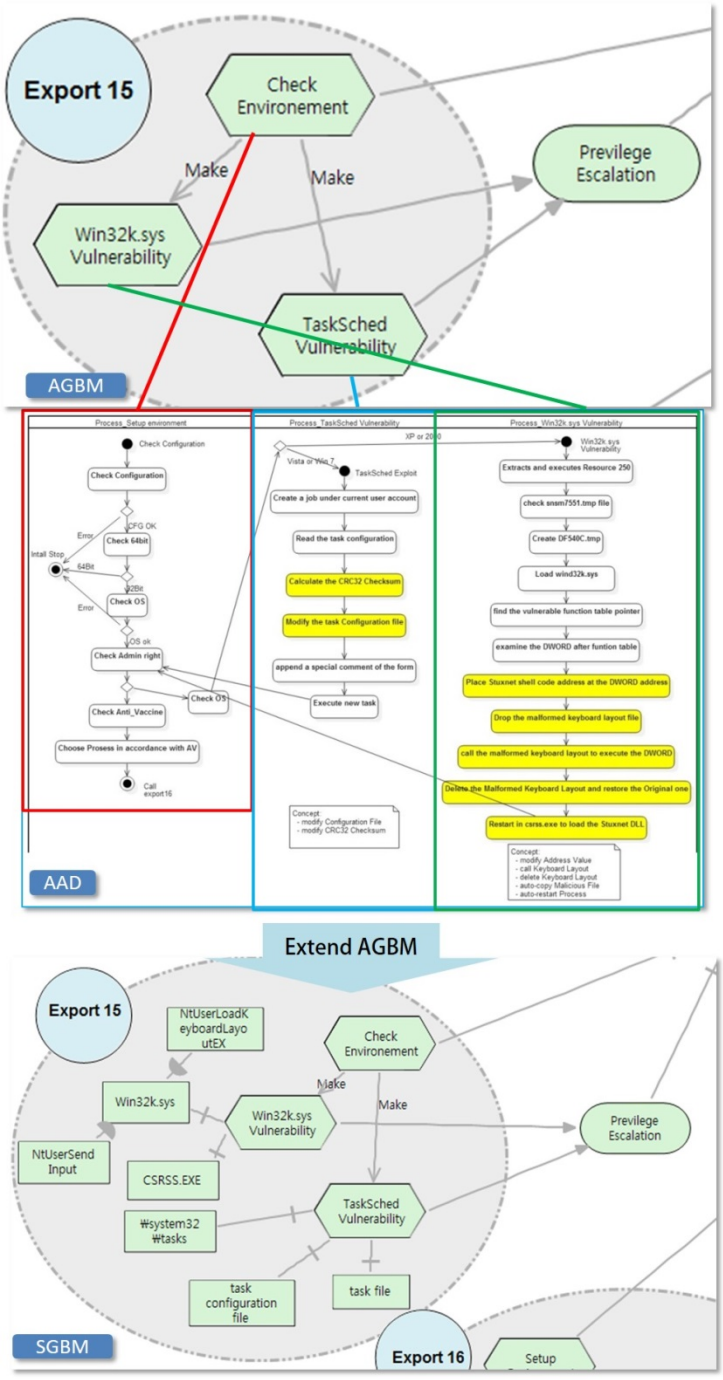

Fig. 4. How to make relationships between AGBM, AAD, and SGBM

4.1 Elicitation, Understanding and Structuring of Stuxnet

We identify what goals and actors are, and what kinds of behaviors are performed. In case of Export 15, it shapes the environment for attack and exploits two types of Zero-day Vulnerabilities based on the Operation System. One is an Exploitation Win32k.sys Vulnerability and the other is an Exploitation Task Schedule Vulnerability. Through the exploitation of these vulnerabilities, Stuxnet can get the Admin Right and call the Export 16.

4.2 Attack Goal-Based Model of Stuxnet (AGBM)

Based on the previous work, we conduct Goal Based Modeling for Stuxnet, and represent goals, tasks, and actors in Fig. 4. We conduct mapping between tasks and goals with And-decomposition and Or-decomposition based on [14], which can be divided into three phases and sub stages: Compromise, Lateral Movement, and Objective Phase. Moreover, each task is included into the related actor. Export 15 includes three tasks: Check Environment, Exploit Win32k sys Vulnerability, and Exploit Task Schedule Vulnerability. Two Exploitation Vulnerabilities are related to the Privilege Escalation.

4.3 Attack Activity Diagram of Stuxnet (AAD)

In case of AAD, we conduct modeling on detailed activities to perform each task in a temporal-ordered manner and apprehend the related vulnerability and access resources. Each vulnerability exploitation is decided based on the OS version. The identified task from AGBM is represented by flows of activities or access of resources, and we can conceptualize the context or describe the resource in order to map them into the SGBM.

4.4 Security Goal-Based Model of Stuxnet (SGBM)

As shown in Fig. 4, resources and vulnerability concepts from AADs are mapped into the AGBM to generate the SGBM. The final SGBM is comprised of Goals, Tasks, Actors, Resources, and Concepts. In the case of the Export 15, Win32k.sys Vulnerability Exploitation out of two exploitations in Privilege Escalation accesses the CSRSS.EXE and Win32k.sys, more specifically the NtUserLoadKeyboardLayoutEx and NtUserSendInput Function related to CVE-2010-2549. In the case of the Task Schedule Vulnerability Exploitation, it needs to access to the Task file, \system32\tasksfolder, and Task Configuration File related to CVE-2010-3338.

4.5 Security Ontology of Stuxent

Using the SGBM, we generate the security requirements ontology based on the Meta-Data Model. The instance in SGBM becomes the subclass of each of the Security Concepts, and each subclass makes a relationship with a related instance. This work provides the inference of related classes. We use the OntoGraf with the basic plug-in in Protege 4.3 for visualization of all related classes from the specific class with all relationships, and this is a strong point of comparison with other visualization plug-ins.

In Fig. 5, we show one example of the visualization about the Exploit Task Schedule Vulnerability and related Security Concepts. First, this task is used to get the Admin Right as the Privilege Escalation. Then, we get the specification including information in previous section: for example, the threat name is Stuxnet, version information is following the detected date, 20100719, the Actor is Export 15, the related CVE is CVE-2010-3338, and the related platform is Windows Vista and the Server 2008 Series.

Though this example just shows only one case, it will become very powerful after gathering knowledge from many cases. As we provided the case study, we understand inferred ideas from the specific concept through the ontology. If we extend the ontology with the aid of knowledge base, it provides various understandings of the context of the situation. Moreover, extending the ontology with the countermeasure, we expect to provide the countermeasure priority and the risk assessment.

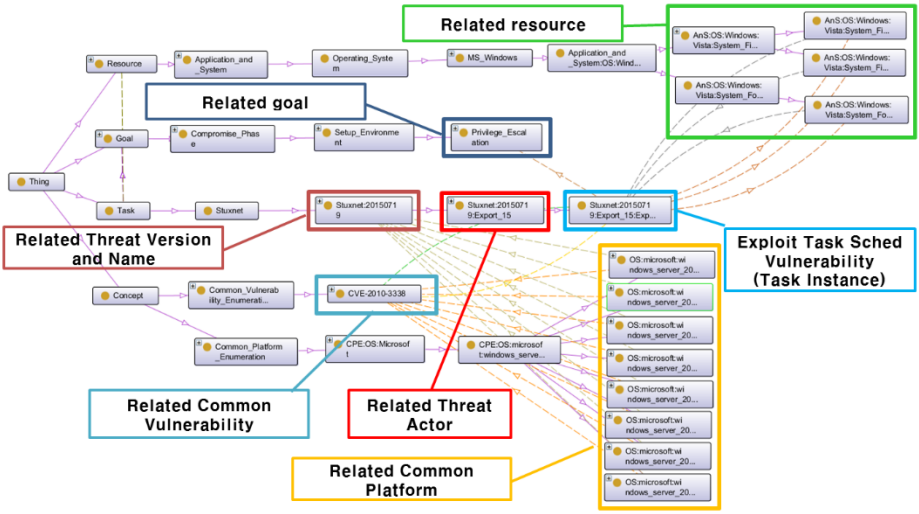

Fig. 5. The visualization for related classes of 'Exploit Task Schedule Vulnerability' using OntoGraf

5 Conclusion and Future Work

Until now, we have discussed the conceptual framework, implemented into the case study using Stuxnet, and have shown how it works. Through this process, we represent the specifications and visualizations of security concepts, and propose the ontology based on the goal-based model. We expect that our research provides understanding of the problem domain in the security requirements in perspective of the requirements engineering process, and generating requirements.

However, as this is the first step of our preliminary research, we need to take further steps, such as extending the ontology including countermeasure and quality attributes, and conducting the research on the detection based on similarities against variant threats and the risk assessment using the security ontology in order to consoli-

date our framework. Ultimately, our final goal is to develop a solid framework based on these kinds of work in order to generate security requirements.

Acknowledgments. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2013R1A1A2009801).

References

1. Virvilis, N., Gritzalis, D.: The Big Four- What we did wrong in Advanced Persistent Threat Detection?. Athens Univ. of Economics and Business, Int'l Conference on Availability, Reliability and Security (2013)
2. Kordy, B., Pietre-Cambacedes, L., Schweitzer, P.: DAG-Based Attack and Defense Modeling: Don't Miss the Forest for the Attack Tree. Univ. of Luxembourg (2013)
3. Elahi, G., Yu, E., Zannone, N.: A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations. Univ. of Toronto, Int'l Conference on Conceptual Modeling (2009)
4. Elahi, G., Yu, E.: A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. Univ. of Toronto, Conceptual Modeling ER (2007)
5. Lin, L., Yu, E.: Security and Privacy Requirement Analysis within a social setting. Univ. of Toronto, Requirement Conference (2003)
6. Li, W., Tian, S.: An Ontology-based intrusion alerts correlation system, Jiaotong Univ. Expert Systems with Application (2010)
7. Wang, J.A., Guo, M.: OVM: An Ontology for Vulnerability Management. Southern Polytechnic State Univ., CSIIRW (2009)
8. Kotenko, I., Polubelova, O., Senko, I.: The Ontological Approach for SIEM Data Repository Implementation. St. Petersburg Instituet for Informatics and Automation(SPIIRAS), IEEE ICGCC (2012)
9. Lee, S.W., Gandhi, R.A., Ahn, G.J.: Certification Process Artifacts Defined as Measureable Units for Software Assurance. The Univ. of North Carolina at Charlotte, Software Process Improvement and Practice (2007)
10. Falliere, N., Murchu, L.O., Chien, E.: W32.Stuxnet Dossier, Symantec, Security Response (2011)
11. Matrosov, A., Rodionov, E., Harley, D., Malcho, J.: Stuxnet Under the Microscope, ESET (2010)
12. Kriaa, S., Bouissou, M., Pietre-Cambacedes, L.: Modeling the Stuxnet Attack with BDMP: Towards More Formal Risk Assessments
13. Ross, R., Oren, J.C., McEvilley, M.: System Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems. National Institute of Standards and Technology, NIST Special publication 800–160 (2014)
14. Bryant, B.: A Method for Implementing Intention-Based Attack Ontologies with SIEM Software. Fishnet Security, Securely Enabling Business White Paper (2014)
15. Horridge, M.: A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3, the University Of Manchester (2011)