Security Ontology: Simulating Threats to Corporate Assets

Andreas Ekelhart, Stefan Fenz, Markus D. Klemen, and Edgar R. Weippl

Secure Business Austria — Security Research,
Favoritenstraße 16, A-1040 Vienna, Austria
{aekelhart,sfenz,mklemen}@securityresearch.at
weippl@ifs.tuwien.ac.at
http://www.securityresearch.at

Abstract. Threat analysis and mitigation, both essential for corporate security, are time consuming, complex and demand expert knowledge. We present an approach for simulating threats to corporate assets, taking the entire infrastructure into account. Using this approach effective countermeasures and their costs can be calculated quickly without expert knowledge and a subsequent security decisions will be based on objective criteria. The ontology used for the simulation is based on Landwehr's [ALRL04] taxonomy of computer security and dependability.

Key words: security ontology, threat modeling, risk analysis

1 Introduction

Over the years, IT-security has become a much diversified field of research, no longer limited to the classical virus attack. Applied IT-security also has to consider physical attacks, acts of nature beyond human control, industrial espionage, etc.

Although security is crucial for every company, approaches to security, including evaluation and implementation of safeguards, vary widely. Wrong decisions are made based on insufficient knowledge about the security domain, threats, possible countermeasures and the own infrastructure. The following reasons have been identified:

First, security terminology is vaguely defined; this leads to confusion among experts as well as the people who should be counseled and served [Don03]. Without a shared terminology communication, especially in a complicated domain like security, cannot be successful. Ontologies provide a perfect solution: not only can terms be defined, but also relationships between them. Reasoning about the generated knowledge opens further possibilities. Nonetheless, at the present time, ontologies are not yet widely used in commercial applications.

Second, the development of adequate security concepts and plans requires a thorough threat analysis. Small and medium sized businesses often avoid this step due to a lack of skilled personnel and budget constraints. Various IT-standards exist: the GSHB [BSI04], for example, is a comprehensive and well

developed approach to security. Companies can become certified if they meet a specified baseline. Drawbacks of this standard are the complexity (approximately 3,000 pages) and insufficient risk analysis support - quantitative risk evaluation is not addressed. Cobit [COB06], an IT governance framework based on best practices, is complex to use and does not address security threats and safeguards in detail. The ISO 17799 standard [ISO06] includes risk analysis and benchmarking, but the operational expense to obtain a certification is usually too high for small and medium sized businesses.

Every security decision must consider the concrete company environment. The employee responsible for security is often not aware of all relevant details of the infrastructure. To obtain the necessary information by interviewing colleagues or studying plans is time consuming and thus the risk analysis is often based on an incomplete picture.

Throughout this paper we use the following definition for the term ontology:

'An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.' [GPFLC04]

Furthermore, we distinguish lightweight and heavyweight ontologies. Lightweight ontologies include concepts, concept taxonomies, relationships between concepts, and properties that describe concepts [GPFLC04]. Heavyweight ontologies add axioms and constraints to lightweight ontologies [GPFLC04].

The security ontology is based on Landwehr's [ALRL04,LBMC94] taxonomy; we extended it to form a heavyweight ontology. Landwehr's security and dependability classification [ALRL04] (for further details see Section "Security Ontology") was enriched by domain specific concepts and attributes to incorporate enterprise infrastructure and role schemes.

The main goal of current research activities is to provide a security ontology that unifies existing approaches such as [BSI04], [ALRL04], [LBMC94], and [eCl06] to support small and medium sized businesses' IT-security risk analysis and mitigation. The ontology "knows" which threats endanger which assets and which countermeasures could lower the probability of occurrence, the potential loss or the speed of propagation for cascading failures. In addition, each infrastructure object in the ontology can be annotated with outage costs and each countermeasure object with acquisition and maintenance costs. These capabilities enable the company to calculate the outage costs caused by a certain disaster and the costs of different countermeasures. The costs of countermeasures can be read directly from the ontology, while the benefit is calculated in simulation runs (see 3.3).

2 Security Ontology

The security ontology consists of three parts. The first part is based on the security and dependability taxonomy by Landwehr [ALRL04], the second part

describes concepts of the (IT) infrastructure domain, and the third part provides enterprises with the option to map their persons and role models.

The ontology is coded in OWL (Web Ontology Language [OWL04]) and the Protege Ontology Editor [Pro05] was used to edit and visualize the ontology and its corresponding instances.

The following subsections describe the parts in more detail:

2.1 Security and dependability taxonomy

Figure 1 shows the *security and dependability taxonomy's* concept structure; for further information the paper [ALRL04] provides the reader with a detailed description. As Figure 1 shows, the taxonomy is designed in a very general

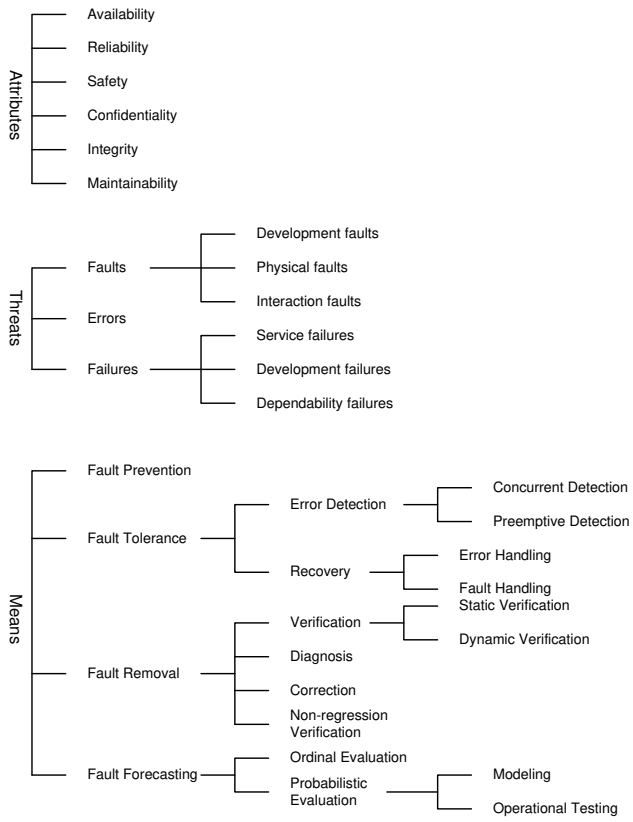

Fig. 1. Security and Dependability Taxonomy [ALRL04]

way, which makes it easy to extend it with additional concepts. Instances and dependencies were inserted to populate the taxonomy. Figure 2 represents the

sub-tree "sec:Threat," which is part of the security ontology and was derived from Landwehr's taxonomy [ALRL04].

The sub-concept "sec:Fire" was inserted as the first real concept; it is classified as a physical fault, which belongs to the super-concept "sec:Threat." With "sec:threatened" and "sec:affects," the first dependencies were inserted; "sec:threatened" describes how every instance of any "sec:Threat" sub-concept threatens all instances of any "ent:Infrastructure" subclass. "sec:affects" shows how every instance of a "sec:Threat" subconcept affects one or more instances of the "sec:Attribute" concept. To every introduced concept in the ontology, a

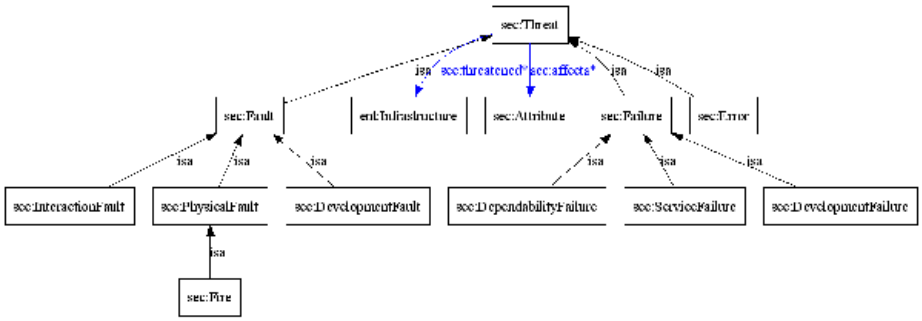

Fig. 2. Sub-tree 'sec:Threat'

semantic definition in natural language in form of a 'rdfs:comment' is added. To provide useful knowledge for simulating threats to corporate assets, the ontology still has to be extended with additional concepts describing the (IT) infrastructure and personnel structure.

2.2 Infrastructure concepts

Figure 3 shows the security ontology's infrastructure area. The company building, with its corresponding floors and rooms, can be described using the infrastructure framework. To map the entire building plan exactly on the security ontology, each room is described by its position within the building. The ontology "knows" in which building and on which floor a certain room is located. The attributes "ent:nextToRoomHorizontal" and "ent:nextToRoomVertical" describe the exact location of each room. Each instance of "ent:ElectronicDevice" and "ent:Safguard" is located in a particular room. A room can, of course, also contain more concepts. The current ontology uses a flexible and easily extendable structure: additional concepts can be included without effort.

The concept "ent:Safeguard" is subdivided into "ent:CounterMeasure" and "ent:Detector," which are used to model detectors (fire, smoke, noise, etc.) and their corresponding countermeasures (fire extinguisher, alarm system, etc.).

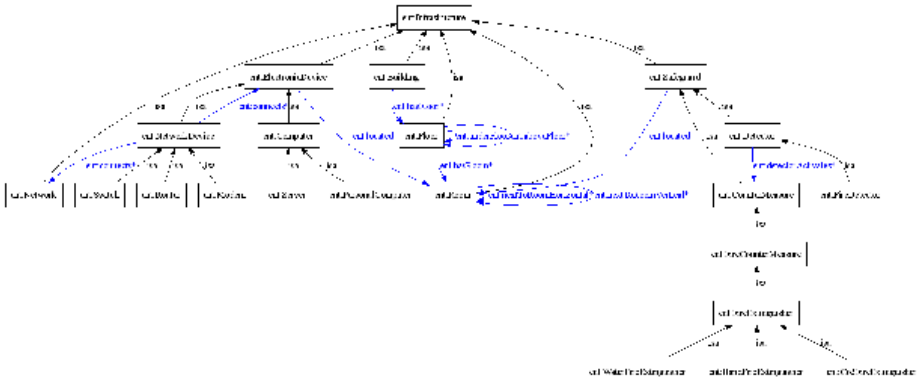

Fig. 3. Sub-tree 'ent:Infrastructure'

Beside "ent:Infrastructure," the concepts "ent:Role" and "per:Person" ensure that both technical and personnel structures can be mapped into the current ontology.

2.3 Person and role concepts

The concept "per:Person" enables the ontology to map natural persons. Figure 4 represents the role concept for assigning certain roles to natural persons. Several

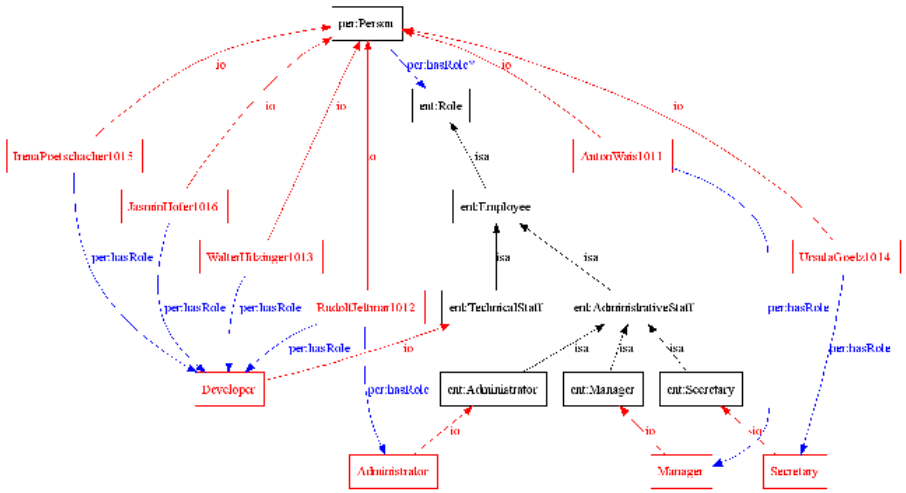

Fig. 4. Sub-tree 'per:Person' and 'ent:Role'

instances of "per:Person" were created in order to assign them with different

roles. The current ontology considers only sub-concepts of "ent:Employee"; if necessary, additional roles can be added easily.

After describing the security ontology, the next section presents a practical example and makes the benefit for SME's clear.

3 Example

In this section we provide an example of how a company would use the aforementioned security ontology to model an IT infrastructure.

3.1 The Company

The company is an SME with six employees. Their main business is software sales and custom programming to modify their standard software. The company rents two floors (first and second floor) of a five-floor building in the center of a small town. On the first floor (Figure 5), there is one office, a storage room and

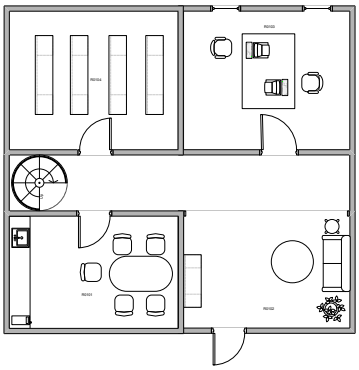

Fig. 5. Floor I

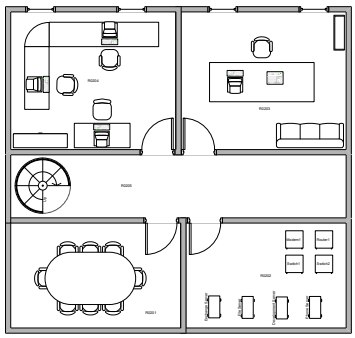

Fig. 6. Floor II

a lunchroom. The server room, a meeting room, and two more offices are located on the second floor (see Figure 6). The following listing shows the allocation of relevant (IT) infrastructure elements:

- First floor - Office room (R0103): 2 PC's
- First floor - Storage room (R0104): data media (archived)
- Second floor - Server room (R0202): 4 Servers, 1 Router, 2 Switches, 1 Modem
- Second floor - Office room (R0203): 1 PC, 1 Notebook
- Second floor - Office room (R0204): 3 PC's

The infrastructure is mapped on the sub-tree "ent:Infrastructure" (compare Figure 3). The following listing gives an example for an OWL definition describing a certain PC with its attributes:

```
<ent:PersonalComputer rdf:ID="Pc4">
 <ent:deliveryTime rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">3</ent:deliveryTime>
 <ent:assetCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">1500</ent:assetCost>
 <ent:outageCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">0</ent:outageCost>
 <ent:antiVirus rdf:datatype="http://www.w3.org/2001/
 XMLSchema#boolean">false</ent:antiVirus>
 <ent:hasOs rdf:datatype="http://www.w3.org/2001/
 XMLSchema#string">WinXpPro</ent:hasOs>
 <ent:located rdf:resource="#R0204"/>
</ent:PersonalComputer>
```

The concept "ent:PersonalComputer" with its concrete instance "Pc4" has the attributes "ent:deliveryTime," "ent:assetCost," "ent:outageCost," "ent:antivirus," "ent:hasOs," and "ent:located." If this or any other instance is destroyed by a particular disaster, the ontology "knows" how long it will take to get a new one, how much it costs, where it is located, and the outage costs per day. Apart from "ent:antivirus" and "ent:hasOs," all attributes are inherited from the supra-concept "ent:Infrastructure."

3.2 The Disaster

After describing the company with its infrastructure, the current subsection defines the disaster that will hit our software company.

We chose the event of fire as a physical threat scenario. The simulation should show the amount of damage over a certain period of time and in consideration of the fire source. A certain room can be defined as the fire source; the speed of propagation without any countermeasures will be five minutes per floor and five minutes per room. Every element of the infrastructure is assigned to a certain room. In the case of fire, all infrastructure elements within a room will be completely destroyed. The outage costs per room correspond to the outage costs sum of all destroyed elements, which are located in the room. It is possible to assign countermeasures to any room. These safeguards can lower the probability of occurrence and the speed of propagation in the case of fire. The attribute "ent:damage" addresses the damage which results when the countermeasure is executed.

The following OWL code-snippet shows an example of the countermeasure element "ent:WaterFireExtinguisher":

```
<ent:WaterFireExtinguisher rdf:ID="WaterFireExtinguisher0102">
 <ent:located rdf:resource="#R0102"/>
 <ent:assetCost rdf:datatype="http://www.w3.org/2001/
 XMLSchema#int">500</ent:assetCost>
 <ent:damage rdf:datatype="http://www.w3.org/2001/
 XMLSchema#float">0.7</ent:damage>
```

```
<ent:deliveryTime rdf:datatype="http://www.w3.org/2001/
XMLSchema#int">5</ent:deliveryTime>
<ent:startTime rdf:datatype="http://www.w3.org/2001/
XMLSchema#float">0.0</ent:startTime>
<ent:extinguishingTimeRoom rdf:datatype="http://
www.w3.org/2001/XMLSchema#int">1</ent:extinguishingTimeRoom>
</ent:WaterFireExtinguisher>
```

We can see that this extinguisher is located in room R0102 and will start immediately when switched on. Instance "WaterFireExtinguisher0102" will extinguish the room within one minute. The attribute "ent:startTime" is important for countermeasures that are not activated automatically (e.g. hand fire extinguisher).

3.3 The Simulation

The framework for our threat analysis has been explained in the preceding sections, we will now present a tool called "SecOntManger," which processes the ontology knowledge of simulated threats. This prototype handles IT costs and poses as proof of concept. Further threat effects, as well as infrastructure components, can be added easily due to the generic structure.

In our example the management wants to know the impact that fire will have on the infrastructure, what countermeasures exist and the benefits they offer. For this purpose we show two program runs, one against the unprotected company and another including safeguards:

- The first program run without countermeasures: "SecOntManager" offers an intuitive graphical user interface, shown in Figure 7. A threat and a corresponding starting point have to be chosen before a simulation can be started. We chose fire as the threat and the server room (room0202) as the origin of the fire. The program run produces a detailed log file, which shows how the fire spreads from room to room and what damage it causes. An abridgment of this file can be seen in the following listing:

```
Current Room: <http://secont.com/secont.rdf#R0203>
        http://secont.com/secont.rdf#Pc7: 0
        used by Person: http://secont.com/secont.rdf#AntonWais1011
        Salary: 3000
        Total outage costs of infrastructure component / 5min: 0.347
        Total damage costs of infrastructure component: 2000
        Recovery time and costs: 4 days: 2200
        ...
Search Detectors:
        Detector found: /
        Countermeasure activated: /
```

  At the end of the simulation all occuring costs are visualized in a line chart (see Figure 7). The time axis unit is set to minutes. Four curves, reflecting different cost categories, exist: The blue curve visualizes the damage. In the example, the damage costs rise very fast due to the speed of fire - within 30

minutes every room was destroyed. By zooming in, displaying only the first 30 minutes, we can see how the damage evolves. After all rooms "burned down," no further damage could occur. Red are the outage costs, taken from assigned outage costs of infrastructure components and employee's costs. Outage costs rise constantly in the simulation until recovery. The green curve shows recovery costs: delivery times for destroyed components are taken into consideration. When components are available and paid, connected outage costs decrease, visually spoken, the red line flattens. Additional installation costs lift the recovery costs upon damage. When every component is recovered, the pre-threat state is reached and outage costs do not rise anymore. Furthermore the total of all costs is reflected by the yellow curve. Fire costs 73,605 and it takes a minimum of five days to recover from the effects.

Fig. 7. SecOntManager: Without countermeasures

– Second program run with countermeasures enabled: We now concentrate on reducing the damage by installing safeguards. "SecOntManager" offers to install fire suppression systems in the building. We decide for pre-action pipes in the entire building. Necessary detectors and fire extinguishers are added to rooms in the OWL file. The costs amount to 7,200. Running the simulation produces the cost chart in Figure 8. As can be seen, the total damage decreases drastically to 28,329. After installing safeguards, the fire cannot spread; it is detected and extinguished shortly after breakout. Nevertheless

costs and recovery times are still very high. The reason is that water extinguishers have a high damage factor concerning electronic devices and we have chosen the server room as place of fire origin. "SecOntManager" also offers CO2 fire extinguishers for locations with high electronically damages. Replacing the water extinguisher by a more expensive C02 extinguisher the total costs are reduced to 10934, which are mostly outage costs of one server which caused the fire. By adding a redundant server the outage time and costs could be cut to zero.

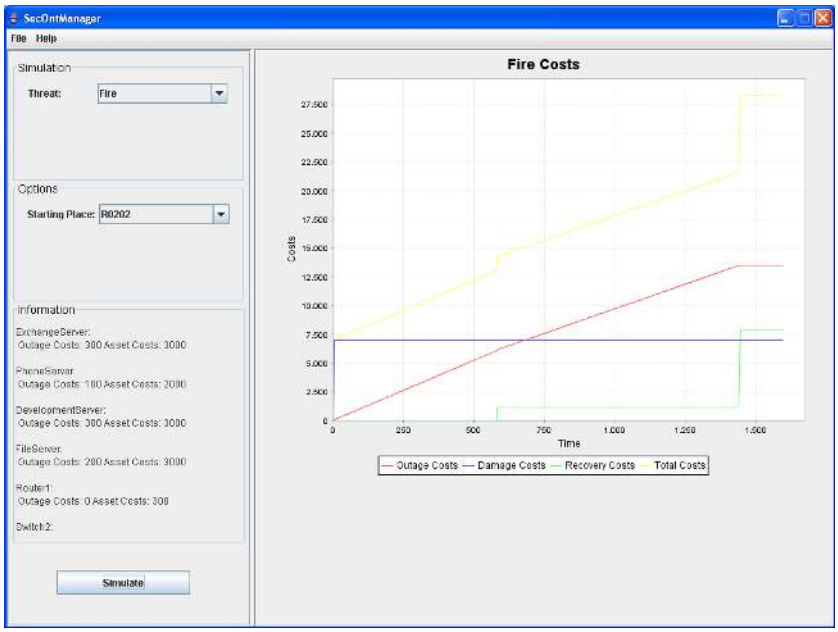

Fig. 8. SecOntManager: With countermeasures enabled

4 Conclusion

We presented an approach that eliminates the former flaws and allows us to simulate threats to corporate assets while taking the entire infrastructure into account. Increasingly, businesses require accurate security concepts and plans to protect themselves and their clients against various threats, including physical attacks, acts of nature beyond human control, industrial espionage, etc. Establishing an all-encompassing IT-security concept demands in-depth knowledge of existing threats, the company, and possible countermeasures. We propose an ontology-based approach combining security- with business-domain knowledge to model companies. The ontology guarantees shared and accurate terminology

as well as portability. Knowledge of threats and corresponding countermeasures, derived from IT-security standards, are integrated into the ontology framework. Moreover, we implemented a prototype capable of simulating threats against the modeled company by processing the knowledge contained in the ontology. "SecOntManager" visualizes the damage caused by specific threats, outage costs, and the recovery time. Running the program with added safeguards shows their benefits and offers objective data for decision making, which safeguards to implement and to avoid installing countermeasures that are not cost-effective. An enhanced prototype with advanced risk analysis and broader threat support will take failure probability into account and will be developed in pilot installations with partner companies.

5 Acknowledgements

This work was performed at the Research Center Secure Business Austria in association with the Vienna University of Technology.

References

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Sec. Comput.*, 1(1):11–33, 2004.

[BSI04] It-grundschutzhandbuch. http://www.bsi.de/gshb/deutsch/download/GSHB2004.pdf, 2004.

[COB06] Cobit. http://www.isaca.org/, 2006.

[Don03] Marc Donner. Toward a security ontology. *IEEE Security and Privacy*, 1(3):6–7, May/June 2003.

[eCl06] eclass. http://www.eclass.de/, 2006.

[GPFLC04] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering*. Springer, London, first edition, 2004.

[ISO06] Iso17799. http://www.iso.org/, 2006.

[LBMC94] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi. A taxonomy of computer program security flaws. *ACM Comput. Surv.*, 26(3):211–254, 1994.

[OWL04] Owl web ontology language. http://www.w3.org/TR/owl-features/, 2004.

[Pro05] The protege ontology editor and knowledge acquisition system. http://protege.stanford.edu/, 2005.