Ontology-based Generation of IT-Security Metrics

Stefan Fenz
Vienna University of Technology - Institute for Software Technology and Interactive Systems
Secure Business Austria
Favoritenstrasse 9-11/E188
Vienna, Austria
+43 1 5031280
fenz@ifs.tuwien.ac.at

ABSTRACT

Legal regulations and industry standards require organizations to measure and maintain a specified IT-security level. Although several IT-security metrics approaches have been developed, a methodology for automatically generating ISO 27001-based IT-security metrics based on concrete organization-specific control implementation knowledge is missing. Based on the security ontology by Fenz et al., including information security domain knowledge and the necessary structures to incorporate organization-specific facts into the ontology, this paper proposes a methodology for automatically generating ISO 27001-based IT-security metrics. The conducted validation has shown that the research results are a first step towards increasing the degree of automation in the field of IT-security metrics. Using the introduced methodology, organizations are enabled to evaluate their compliance with information security standards, and to evaluate control implementations' effectiveness at the same time.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; K.6.4 [**Management of Computing and Information Systems**]: System Management

General Terms

Security, Management, Measurement, Economics.

Keywords

Metrics, Security, Compliance, Economics.

1. INTRODUCTION

With the extensive use of information technology in all business areas the requirements on IT-security widened dramatically [8, 2]. Additionally, legal regulations and industry standards force organizations to adopt a holistic security approach. Information security standards such as ISO 27001 [6] and best-practice guidelines such as the German IT Grundschutz Manual [3] are an excellent way for providing organizations with concrete IT-security knowledge to maintain or enhance their overall IT-security level. The main problem of these standards and best-practice guidelines is that they only define controls and potential control implementations and do not allow to make statements about their qualities like effectiveness. However, for a holistic security program it is indispensable to continuously evaluate the compliance regarding information security standards and the effectiveness of already existing control implementations [9]. IT-security metrics are a potential technique to analyze and monitor the employed control implementations (cf. [4] for a detailed IT-security metrics definition).

While IT-security metric standards such as NIST SP 800-55 [4] provide detailed information about how to create, manage, and apply IT-security metrics in general, they do not give detailed guidance regarding the objects of measurement and the way how to obtain the necessary input data for the calculation. The main problem of existing contributions in the field of IT-security metrics (e.g. [10, 11, 1]) is that they do not provide a solution of how to derive the knowledge which is necessary for calculating IT-security metrics (e.g. concrete control compliance requirements, existing control implementations in the organization including their effectiveness, etc.).

What we need is a formal information security knowledge base (security ontology) which is capable of storing and interrelating general information security knowledge and organization-specific knowledge about existing control implementations to automatically generate comprehensible IT-security metrics. But why do we use ontologies and not a simple database solution to store the knowledge necessary for the metrics generation? First, we want to be able to define the entire domain in a standardized way (the security ontology is modeled with the W3C ontology language OWL). Second, we want to use description logics and reasoning engines to relate the general with the organization-specific part of the security ontology (e.g. determining if a specific resource is compliant to a specific ISO 27001 control in a specific setting of an organization). Third, using reasoning engines and ontology languages relying on accepted standards (e.g. Pellet and OWL) allows us to skip the development of our own description logics engines. Thus and in the context of a specific security ontology, the overall research question of this paper is:

SAC'10 March 22-26, 2010, Sierre, Switzerland.

- How can the security ontology by Fenz et al. and existing reasoning engines be used to support the generation and maintenance of ISO 27001-based IT-security metrics?

Based on the security ontology by Fenz et al. (cf. [5]), including information security domain knowledge and the necessary structures to incorporate organization-specific assets into the ontology, this paper proposes a methodology for automatically creating ISO 27001-based IT-security metrics. The following section provides a brief overview of the security ontology which is used as the basis for the IT-security metrics generation.

2. SECURITY ONTOLOGY

Figure 1 shows the high-level concepts and corresponding relations of the security ontology (see [5] for a detailed description of the concepts and relations). In the context of this paper only the most relevant concepts Control and Asset are described in detail. Each control is implemented as asset concept, or as combinations thereof. Controls are derived from and correspond to best-practice and information security standard controls (e.g. the German IT Grundschutz Manual [3] and ISO/IEC 27001 [6]) to ensure the incorporation of widely accepted knowledge. Each security ontology control contains a formal description that defines under what circumstances the control is fulfilled - currently four possible control implementation contexts exist: (1) organizational context - an organization is compliant to control x if control implementation combination y is implemented at the organization (e.g. organization-wide data back up policy), (2) technical context - a physical environment is compliant to control x if control implementation combination y is implemented at the physical environment (e.g. fire extinguishing system and smoke detectors in a building), (3) software context - a computer is compliant to control x if control implementation combination y is implemented at the computer (e.g. single anti virus solution or entire internet security suite), and (4) network context - a network is compliant to control x if control implementation combination y is implemented at the network (e.g. intrusion detection system and fire wall). When implementing the controls, compliance with various information security standards is implicit. To enrich the ontology with concrete information security knowledge (threats, vulnerabilities, and formal control descriptions) parts of the German IT Grundschutz Manual have been superimposed on the security ontology. The coded ontology follows the OWL-DL (W3C Web Ontology Language) [12] standard and ensures that the knowledge is represented in a standardized and formal form. Each high-level concept depicted in Figure 1 summarizes several sub-concepts. Currently, the security ontology contains 512 primitive concepts, 119 defined concepts, and 36 relation types. To utilize the ontology and its inference capabilities for automated compliance checks and the IT-security metrics generation the organization has to model their technical and organizational environment as OWL individuals according to the defined concept model (cf. Section 3).

3. ONTOLOGY-BASED GENERATION OF IT-SECURITY METRICS

International and national institutions such as ISO and the U.S. National Institute for Standards and Technology

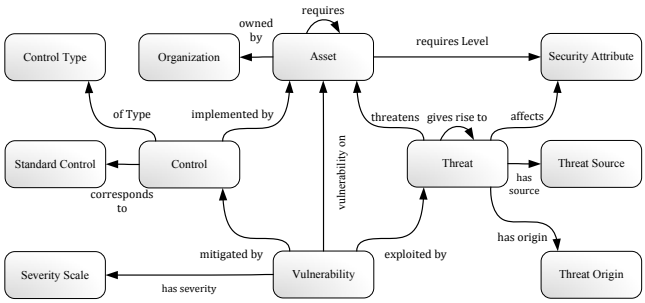

Figure 1: Security Ontology Top-Level Concepts and Relationships

(NIST) are developing IT-security metric standards which give instructions on the development and use of measures in order to evaluate an organization's IT-security level based on controls and objectives provided by an information security standard. Based on the controls of the used information security standard the generation of IT-security metrics requires:

- Exact definition of the scope of measurement
- Specification of the compliance requirements
- Information on the quality (e.g. effectiveness) of existing control implementations

In this paper we show how we utilize the security ontology and reasoning engines to generate ISO 27001-based IT-security metrics incorporating organization-specific control implementation knowledge. Figure 2 gives an overview of the proposed methodology and describes how we use the security ontology by Fenz et al. and existing reasoning engines to support the generation and maintenance of ISO 27001-based IT-security metrics (original research question).

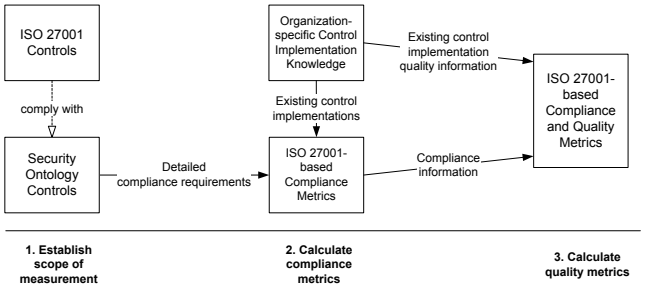

Figure 2: Methodology for Creating ISO 27001-based IT-Security Metrics

1. **Establish scope of measurement:** In the first step we use ISO 27001 controls, which demand for the fulfillment of specific IT-security requirements. Since each security ontology control is related to one or more ISO 27001 controls we use the formal security ontology controls and their compliance requirements to establish the scope of measurement.
2. **Calculate compliance metrics:** Based on the compliance requirements we use reasoning engines to extract the knowledge regarding the organization's control implementations from the security ontology and

calculate the degree of compliance (this step assumes that all relevant parts of the organization – e.g. IT infrastructure, building infrastructure, and available policies – are modeled and mapped to our ontological model).

3. **Calculate quality metrics:** In the third step, the generated compliance information from Step 2 is used together with the quality information regarding existing control implementations (extracted from the security ontology) to generate the final ISO 27001-based compliance and quality metrics.

The following subsections provide a detailed description of each step and the prerequisites for conducting the steps of the proposed methodology.

3.1 Prerequisites

First, it is necessary that the organization maps its technical and organizational environment to the security ontology (cf. Section 5 for a discussion regarding the complexity of the mapping process). Currently, we use the Protege ontology editor to model the (1) physical reality (buildings, rooms, windows, doors, building location), (2) organizational reality (existing policies and contracts), and (3) IT-related reality (clients, servers, network hardware, installed software, and entire networks). Each piece of reality is represented by OWL individuals in the existing concept structure of the security ontology. Relations connect the concepts and thus the individuals to each other (e.g. asset_contains_Asset, organization_implements_Policy, itcomponent_connectedTo_ITComponent, or sectionConnector_connects_Section). After modeling and mapping an organization's technical and organizational environment to the security ontology, it offers a wide range of machine-interpretable knowledge concerning an organizations structure and its security issues. Remember: the formal compliance requirements define which assets have to be in place to fulfill a certain control. Therefore, the inventory of the organization's assets enables us to use reasoning engines to determine the organization's degree of compliance. Each security ontology control corresponds to one or more ISO 27001 controls. This ontological relationship enables us to use the security ontology as the basis for the ISO 27001-based IT-security metrics generation.

3.2 Scope of Measurement

ISO 27001 [6] defines the requirements and objectives of information security and groups them into eleven categories. Each of the categories is further divided into several subcategories. E.g. physical and environmental security is subdivided into the categories Secure Areas and Equipment Security. These subcategories and the corresponding high-level compliance requirements are the most detailed definitions provided by ISO 27001. Based on the German IT Grundschutz Manual, ISO 27001, and the BSI IT Grundschutz - ISO 27001/27002 allocation tables[1] the security ontology provides formal descriptions of which assets (control implementations) are required in which context to fulfill a certain ISO 27001 control. The required assets to fulfill a certain ISO 27001 control are derived with the allocation tables from the highly detailed IT Grundschutz controls. For exemplification Table 1 shows the relations between ISO 27001 Control 9.1.1. (Physical Security Perimeter) and the security ontology controls necessary for achieving compliance with ISO 27001 Control 9.1.1. Each security ontology control is equipped with a formal implementation description, i.e., the control is fulfilled if the required asset is implemented in the context of the considered control. Table 1 shows which assets have to be implemented to fulfill ISO 27001 control 9.1.1: site location policy, perimeter protection, access systems, entry checkpoints or security guards, safety doors, and one of the listed security window types. The ISO 27001 control 9.1.1 is fulfilled if these asset types are implemented in the defined context (e.g. organization, site, building, or section). In most cases more than one security ontology control is required to reach compliance with a certain ISO 27001 control. That means each ISO 27001 control demands for its own metric that makes statements about its compliance status and the quality of existing control implementations:

Nr.	ISO Control	Security Ontology Control	IT Grundschutz Control	Required Assets to Fulfill Security Ontology Control
9.1.1	Physical Security Perimeter	SiteLocationControlCompliantOrganization	M 1.16	SiteLocationPolicy
		PerimeterProtectionControlCompliantSite	M 1.55	PerimeterProtection
		AccessRegulationControlCompliantBuilding	M 2.17	AccessSystem
		EntranceControlServiceControlCompliantBuilding	M 1.17	EntryCheckpoint or SecurityGuard
		SafetyDoorsControlCompliantSection	M 1.10	SafetyDoor
		SecureWindowsControlCompliantSection	M 1.10	WiredWindow or AcrylicWindow or SecurityFilmWindow or LaminatedWindow or TemperedWindow

Table 1: Exemplary Relationships among ISO 27001, IT Grundschutz, and Security Ontology Controls

- Compliance Metrics: have the aim to check the existence of control implementations and thereby to check which ISO 27001 controls are currently fulfilled by the considered organization → WHAT controls are already implemented?

- Quality Metrics: check the quality of already existing control implementations → HOW are existing control implementations implemented?

3.3 Compliance Metrics

The compliance metrics are required to show the ISO 27001 compliance of a specific asset (e.g. customer data or mail server). A specific asset is compliant to ISO 27001 when all relevant controls are fulfilled with a rate of 100 percent. A specific environment (e.g. an organization) is compliant to ISO 27001 if all relevant assets are compliant to ISO 27001. According to the proposed methodology each security ontology control connected to an ISO 27001 control has to be fulfilled to achieve ISO 27001 compliance. Therefore, the compliance metrics have to calculate percentages which show the level of fulfillment of a certain security ontol-

[1]Allocation table - ISO 27001/27002 to German IT Grundschutz Manual: `http://www.bsi.de/gshb/deutsch/hilfmi/isovergleich/Vergleich_ISO27001_GS.pdf`, last access: 21 October 2009

ogy control. This can be the percentage of computers protected by anti virus software, the percentage of employees who received security awareness trainings, or the percentage of secure windows in sensitive areas. To systematically represent the compliance metrics we introduce two measures: (1) measure X describes, in the context of the considered security ontology control, the number of required control implementations, and (2) measure Y describes in the same context the number of already existing control implementations. A measurement function Z calculates the degree of compliance (Z = Y/X). If $Z < 1$ the control is not fulfilled, the causes must be identified and corrective measures must be taken. The following procedure has been developed to automatically generate the asset-specific compliance metrics from the security ontology:

1. Identification of the asset that should be protected (e.g. the customer data)

2. As the organizational, physical, and technical environment of the organization has been mapped to the security ontology we use a reasoner to determine in which organizational, physical, and technical environment the asset is embedded (e.g. the asset is owned by organization ACME, is physically located in room Y, is technically located on server X that is connected to the ACME LAN that is connected via a fire wall to the Internet).

3. Each security ontology control formally defines the requirements for a compliant environment in a specific context (i.e., for a compliant (1) organization, (2) section, (3) building, (4) computer, or (5) network). For each security ontology control we use a reasoner to determine the compliance status of each asset which is organizationally, physically, or technically connected to the asset identified in Step 1 (e.g. check if the computer where the customer data is located is compliant to the anti virus control, i.e. to be compliant an anti virus solution has to be installed on this computer).

4. The reasoner returns a list of compliant and non-compliant environments. By utilizing the Protege 4 explanation engine[2] we are able to provide detailed explanations for each environment state.

The following example shows how the procedure is utilized to check the compliance of the customer data asset regarding to the ISO 27001 9.1.1 control. Amongst others the ISO 27001 9.1.1 control is connected by relation *correspondsTo* to the security ontology control *SecureWindowControlCompliantSection*, which is implemented by safety window concepts (e.g. acrylic window, wired window, security film window, laminated window, or tempered window). I.e., any section that exclusively implements safety windows is considered – in the given control context – as secure by the reasoner. Via the relation *asset_locatedIn_Asset* we infer that the customer data asset is located in the server room. Therefore, we have to check if the server room is considered as secure regarding the *SecureWindowControlCompliantSection* control definition:

SecureWindowControlCompliantSection =
Section AND
not(section_connectedBy_SectionConnector some StandardWindow)

Read: any OWL individual that is asserted to concept 'Section' and is not connected to any OWL individual that is asserted to concept 'StandardWindow' by relation section_connectedBy_SectionConnector will be classified as secure section regarding the secure window control. In other words, the reasoner checks if only safety windows are used in the server room (section) where the customer data is located: (1) our solution checks how much potential control implementations exist (e.g. the room has 15 windows → X = 15), and (2) depending on the formal control implementation description of the considered security ontology control the reasoner checks if a valid control implementation exists (e.g. from 15 windows, 13 windows are an instance of a safety window subtype → Y = 13). As only 13 of 15 windows are safety windows, the server room is not classified as a secure window control compliant section. Instead the server room is classified according to the corresponding *SecureWindowControlNoncompliantSection* control definition:

SecureWindowControlNoncompliantSection =
Section AND
section_connectedBy_SectionConnector some StandardWindow

Read: any OWL individual that is asserted to concept 'Section' and is connected to any OWL individual that is asserted to concept 'StandardWindow' by relation section_connectedBy_SectionConnector will be classified as unsecure section regarding the secure window control. We use the noncompliant control definitions to provide causes of deviation at the compliance metrics. In the described example the Protege 4 explanation engine identifies based on the security ontology the standard window instances *StandardWindow_4* and *StandardWindow_7* located in the server room as the cause for non-compliance and communicates this fact to the user. Applying the measurement function from the compliance metrics we calculated the degree of compliance for the secure window control → $Z = Y/X = 13/15 = 0.87$. The result shows that 87 percent of the windows comply to the considered control. The described compliance metrics calculation is conducted for each security ontology control and summarized to the ISO 27001 control level.

3.4 Quality Metrics

In contrast to the compliance metrics the quality metrics are used to determine the effectiveness of existing control implementations (e.g. anti-theft devices, safety windows, or fire-protection installations) in ISO 27001 compliant environments. The security ontology provides for a given security ontology control / control implementation context a qualitative effectiveness rating (high, medium, low). Each item of the qualitative rating scale is connected to a numerical value (high = 3, medium = 2, low = 1). Based on this rating scale it is possible to automatically determine the effectiveness of control implementations and control implementation bundles. To systematically represent the quality metrics we split them into four main parts: (1) base measure X to obtain the total amount of a specific control implementation in a given environment, (2) base measure Y to calculate the sum of their effectiveness values, (3) derived

[2]Protege 4 Explanation: `http://owl.cs.manchester.ac.uk/explanation/`, last access: 21 October 2009

measure specification Z, and (4) specification of indicators to interpret the derived measurement function results. To determine existing control implementations we use the procedure described before to extract the necessary data from the ontology. In contrast to the compliance metrics we also query the effectiveness of existing control implementations in the context of the considered control. The security ontology provides for each security ontology control / control implementation combination an effectiveness rating (high, medium, low). The following example is based on the following assumptions: (1) the compliance check revealed that the server room is exclusively equipped with safety windows, and (2) 15 safety windows exist; 10 windows are wired windows (effectiveness = high = 3); 3 windows are laminated windows (effectiveness = medium = 2); 2 windows are tempered windows (effectiveness = low = 1) $\rightarrow X = 15$, $Y = 10*3+3*2+2*1 = 38$, and $Z = Y/X = 38/15 = 2.53$. The result shows that the average effectiveness of the existing safety windows in the office is 2.53. The interpretation of this result is done by organization- or industry-specific indicator specifications (currently not provided by the security ontology). In our example an average effectiveness level above 2.5 would be satisfactory. Thus, the current control implementations in the context of the considered control comply to the organization's requirements.

4. IMPLEMENTATION

Subsequent to describing the developed approach for the ontology-based generation of IT-security metrics, this section focuses on its prototypical implementation.

Architecture.

The implementation of the developed approach follows the service-oriented architecture paradigm. The client side implements the WPF-based user interface and connects itself to a Java-based web service that hosts the actual metrics engine and associated components (cf. Figure 3). The metrics engine uses the Protege 4 explanation engine, the Pellet reasoner, and the Java-based security ontology service to generate the compliance and quality metrics for the user interface. The security ontology service acts as an interface to the actual OWL-based security ontology.

Collaboration of the components.

The single components are integrated as follows to generate requested IT-security metrics (cf. Figure 4):

1. If the user requests a metrics report regarding a specific context, the user interface invokes the connect method of the metrics engine. The metrics engine loads the actual security ontology at the security ontology service. Invoked by the security ontology service the Pellet reasoner classifies the security ontology and determines according to the formal control definitions the compliance status of each OWL individual representing a resource of the organization.

2. The given metrics context is used by the user interface to trigger a metric request by passing the metric context (e.g. an organization, a department, or a computer) to the metrics engine. The metrics engine invokes the getContextResources method at the security ontology service. By querying the security ontology

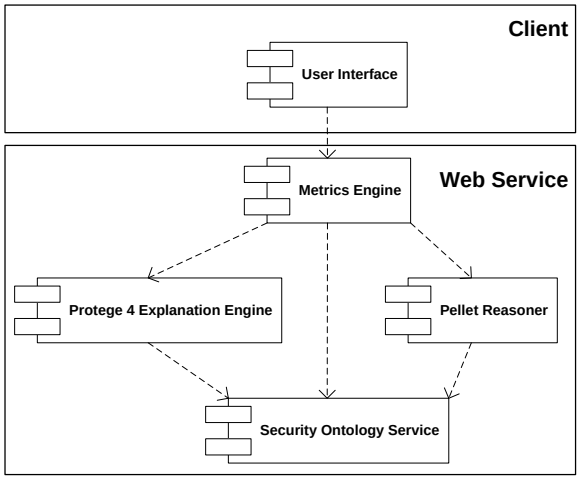

Figure 3: Component View

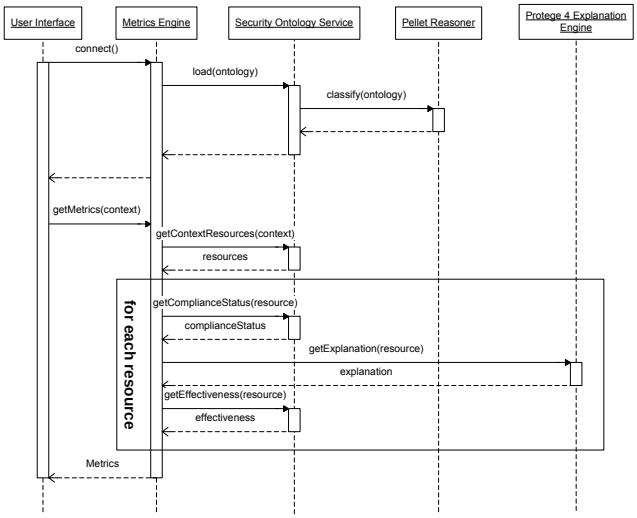

Figure 4: Collaboration of the Components

the service returns all modeled resources which are associated with the metrics context in a technical, organizational, or physical way. E.g. if we want to generate the metrics for a specific server the method would return all OWL individuals representing resources associated with the server (e.g. the organization that owns the server, the room where the server is located or the network that is connected to the server).

3. For each resource that is returned by the getContextResources method the metrics engine invokes the getComplianceStatus method at the security ontology service to check its compliance status. Note: the compliance status has already been determined by the Pellet reasoner in the first step.

4. For each compliant and non-compliant resource the metrics engine invokes the getExplanation method at the Protege 4 explanation engine to get the reasons for the compliance status of the resource.

5. The returned explanations contain resources that protect the given metrics context and therefore fulfill the formal control descriptions (e.g. the server is compliant regarding a certain control because the server room contains an automatic fire extinguishing system). For each resource mentioned in the explanations the metrics engine determines its effectiveness in the given control context to calculate the quality metrics (e.g. the effectiveness of a specific data backup policy).

6. Finally, the metrics engine generates a XML string containing the generated compliance and quality metrics and returns it to the user interface for visualization.

Figure 5 partially shows the final IT-security metrics report containing compliance and quality metrics related to ISO 27001 controls. Please note that this figure shows a proof of concept and does not correctly incorporate all security ontology and ISO 27001 controls.

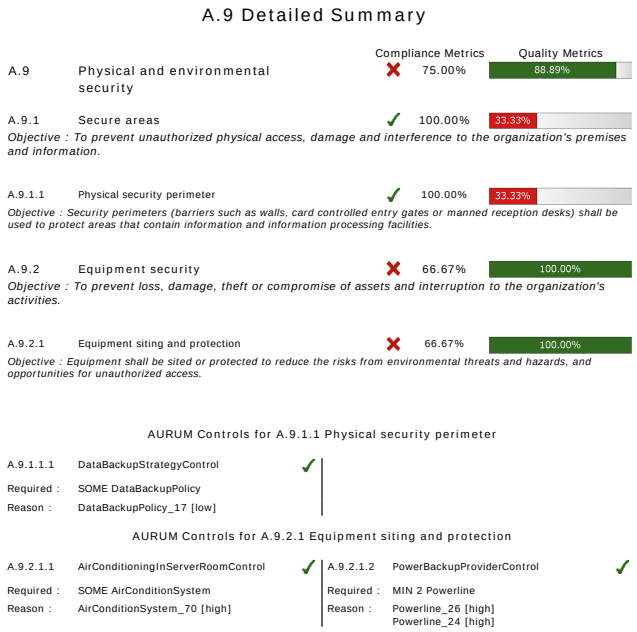

Figure 5: Final IT-Security Metrics Report

5. VALIDATION

To validate the developed ontology-based generation of IT-security metrics, a team of eight information security professionals was compiled to assess the approach and how it utilizes the security ontology to generate compliance and quality metrics in the field of IT-security. Exemplary scenarios, calculations, and result sets have been used at the validation. The subsequent discussion has been dominated by the following issues:

Complexity.

The main problem with this approach is its complexity. It requires an initial mapping of the organization's resources and environment to the ontology and the knowledge body has to be updated if resources or environment change. Although, this seems to be a very complex task we have to consider that every attempt to measure the level of IT-security requires the assessment of existing resources and the relevant environment. The advantage of the ontological solution is that it provides defined concepts and relationships and the organization is only required to model their resources and environment within the given concept structure. This means that the ontology enables organizations to skip the part where they have to think about fundamental IT-security knowledge or concrete control implementation requirements. To enhance the usability of our solution we are currently working on a web-based solution which enables a collaborative and supervised assessment of the resources and the environment. Check-lists based on the control implementation descriptions, guide the responsible persons (e.g. process or information owners) at what the should assess. This enables us to decentralize the assessment step and share the workload. To ease the inventory of computers, network scanning tools are used to automatically capture IP addresses and host names in the given subnet. The time required to assess the necessary information depends on (1) the size of the organization, and (2) the quality of existing documentation.

Weakest Link Problem.

Another limitation has been identified at the quality metric calculation. While for the compliance metrics including the number of control implementations is irrelevant, the quality metric results depend on the amount of considered control implementations. When the number of control implementations included in the calculation is too large, no serious interpretations of the results are possible. For example seven control implementations with effectiveness high (3) and three control implementations with effectiveness low (1) show an average level of security of 2.4. 997 high effective control implementations and 3 low effective control implementations show an average security level of 2.994. This result feigns that the security of existing control implementations is almost perfect, although 3 items are classified with the lowest effectiveness → ignorance of the weakest link. To solve this problem the reporting format should additionally deliver the number of control implementations with the lowest effectiveness occurring in the respective calculation.

Another solution would be to modify the indicator values interpreting a metrics result respective to the number of considered control implementations.

6. RELATED APPROACHES

Although no ontology-based approaches for the generation of IT-security metrics exist, a multiplicity of general approaches for supporting automated metric calculation exist (cf. [7]):

Spreadsheets: Whenever thoughts about introducing an approach for calculating something (e.g. IT-security metrics) are made, the intention to generate a spreadsheet which acts as the data basis and user interface comes up. A spreadsheet allows to implicate many issues concerning security metrics: prototypes of new metrics can be tested, it provides help for evaluating data samples due to their usefulness, data obtained by manual collection methods can be consolidated and a metrics program as a whole can be piloted.

Business Intelligence Tools: Business intelligence and data-mining tools have the ability to obtain data from almost all sources like general-purpose enterprise systems, XML files, relational databases, and flat files, and they allow the definition and calculation of IT-security metrics to observe an organizations security status. Their intention is to provide snapshots of an organization's present situation by providing ad-hoc explorations of data sets.

Security Event and Incident Management Systems: Security event and incident management systems can be adapted for the use as automated IT-security metric solutions. Unfortunately, security event and incident management systems focus on taking corrective actions within minutes or seconds, and do not refer to long running IT-security metrics. These strategic metrics also often need a wider view on an organizations security system whereas security event and incident management systems normally focus on limited scopes.

The described solutions are either not intended for managing IT-security metrics or focus only on special areas but disregard the holistic view of IT-security. The ontology-based generation of IT-security metrics provides organizations with an out-of-the-box tool to check their ISO 27001 compliance. The organization is only required to model and map their organizational and technical environment to the security ontology. The degree of compliance is automatically determined by the included information security knowledge and the corresponding rule-sets.

7. CONCLUSION

Measuring IT-security is one of the grand challenges in IT-security. Although, several IT-security metric approaches have been developed, a methodology for automatically creating concrete IT-security metrics incorporating organization-specific control implementation knowledge is missing. Based on the security ontology by Fenz et al. we proposed a methodology for automatically generating ISO 27001-based IT-security metrics and showed how the security ontology can be used to generate concrete and organization-specific knowledge regarding existing control implementations. On the example of ISO 27001 we showed that the developed methodology supports organizations at evaluating (1) their compliance to information security standards, and (2) the effectiveness of existing control implementations. Further research will address the identified limitations and will strive for the incorporation of further information security and industry standards. We plan to align our ontology-based IT-security metric generation with the ISO 27004 information security management standard, which is expected to be released in fourth quarter of 2009. Furthermore, we will take the evaluation of our concepts from an expert to a real-world level by applying our concepts in real-world audit scenarios. The planned research activities will constitute our second step towards increasing the degree of automation in the field of IT-security metrics.

Acknowledgment

This work was supported by grants of the Austrian Government's FIT-IT Research Initiative on Trust in IT Systems under the contract 813701 and was performed at the research center Secure Business Austria funded by the Federal Ministry of Economy, Family and Youth of the Republic of Austria and the City of Vienna.

8. REFERENCES

[1] M. D. Aime, A. Atzeni, and P. C. Pomi. The risks with security metrics. In *QoP '08: Proceedings of the 4th ACM workshop on Quality of protection*, pages 65–70, New York, NY, USA, 2008. ACM.

[2] BERR. 2008 information security breaches survey. Technical report, Department for Business Enterprise and Regulatory Reform (BERR), April 2008.

[3] BSI. IT Grundschutz Manual, 2004.

[4] E. Chew, M. Swanson, K. Stine, N. Bartol, A. Brown, and W. Robinson. Performance measurement guide for information security. NIST Special Publication 800-55, National Institute of Standards and Technology (NIST), July 2008.

[5] S. Fenz and A. Ekelhart. Formalizing information security knowledge. In *Proceedings of the 4th ACM Symposium on Information, Computer, and Communications Security*, pages 183–194, New York, NY, USA, 2009. ACM. 978-1-60558-394-5.

[6] ISO/IEC. ISO/IEC 27001:2005, Information technology - Security techniques - Information security management systems - Requirements, 2005.

[7] A. Jaquith. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley, 2007.

[8] A. C. Johnston and R. Hale. Improved security trough information security governance. *Communications of the ACM*, 52(1):126–129, January 2009.

[9] K. Julisch. Security compliance: The next frontier in security research. In *NSPW '08: Proceedings of the New Security Paradigms Workshop 2008*, pages 71–74. ACM, 2008.

[10] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, and D. Wright. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, 1993.

[11] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experiments with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, 1999.

[12] W3C. OWL - web ontology language, February 2004.