Towards an Abstraction Layer for Security Assurance Measurements (Invited Paper)

Teemu Kanstrén, Reijo Savola, Antti Evesti, Heimo Pentikäinen
VTT Technical Research Centre of Finland, Oulu, Finland
{firstname.lastname}@vtt.fi

Artur Hecker
Telecom ParisTech
Paris, France
artur.hecker@enst.fr

Moussa Ouedraogo
Research Center Henri Tudor
Kirchberg, Luxembourg
moussa.ouedraogo@tudor.lu

Kimmo Hätönen, Perttu Halonen
Nokia Siemens Networks
Espoo, Finland
{firstname.lastname}@nsn.com

Christophe Blad
Oppida
Montigny le Bretonneux, France
christophe.blad@oppida.fr

Oscar López, Saioa Ros
Nextel S.A.
Zamudio, Spain
{firstname.lastname}@nextel.es

ABSTRACT
Measurement of any complex, operational system is challenging due to the continuous independent evolution of the components. Security risks introduce another dimension of dynamicity, reflected to risk management and security assurance activities. The availability of different measurements and their properties will vary during the overall system lifecycle. To be useful, a measurement framework in this context needs to be able to adapt to both the changes in the target of measurement and in the available measurement infrastructure. In this study, we introduce a taxonomy-based approach for relating the available and attainable measurements to the measurement requirements of security assurance plans by providing an Abstraction Layer that makes it easier to manage these dynamic features. The introduced approach is investigated in terms of a security assurance case example of firewall functionality in a Push E-mail service system.

Categories and Subject Descriptors
D.4.6 [**Security and Protection**]: Invasive Software.

C.4 [**Performance of Systems**]: Measurement Techniques.

General Terms
Measurement, Security.

Keywords
dynamicity, measurement, taxonomy, abstraction

1. INTRODUCTION
In the CELTIC BUGYO Beyond (Building Security Assurance in Open Infrastructures – Beyond) [1] project, work has been carried out on supporting the modeling of security assurance relevant components and services and monitoring their actual implementation against the expectations encoded in the specified security assurance model. In this context, we define security assurance as evaluating that security countermeasures are running as expected (as grounds for confidence that an entity meets its security objectives [2]). In other words, correctness of security controls is the main measurement objective. Our targets are especially systems in the telecommunication domain. In such complex environments, the needs for monitoring will continuously evolve (e.g. due to change in security risk, updated risk management decisions, changes in compliance requirements, or changes in service infrastructure). Similarly, the availability of measurements will evolve over time in different cases, e.g., when new *measurement probes* (or new versions of old probes) become available or existing ones are removed. During some point in time, certain measures can even be delivered by several available probes. A measurement probe is defined here as a tool for performing checks on infrastructure objects in order to provide required information to assess that the deployed security controls are running as expected.

In order to provide means for sharing information about the security assurance requirements for different services, we have defined the concept of Assurance Profile (AP) [3], which is a formalization of the needs for a common set of security assurance measurements on a specified Target of Measurement (ToM), worked out by equipment vendors, solution providers, service integrators, operators and service providers. The ToM is defined as a minimal set of *Infrastructure Objects* where security controls realization is implemented and for which security assurance measures are required.

A Measurement Framework (MFW) used for security assurance of ToM needs to be able to adapt to all above mentioned changes in the ToM, security risk (reflected to security assurance requirements), and to changes in the measurement infrastructure. The framework must be able to interpret the abstract definition of measurement requirements defined in an AP and map them back to available and attainable measurements. In order to make this approach practically useful, the adaptation needs to be performed automatically and with minimal manual intervention. At the same

ECSA 2010, August 23–26, 2010, Copenhagen, Denmark.

time, the correctness and optimal effectiveness of the measurements needs to be ensured in terms of available measures and their properties.

In this paper, we propose a conceptual approach to provide an Abstraction Layer between the measurements provided by the measurement infrastructure and the measurement requirements presented to the measurement framework by its user. We discuss how this allows the measurement requirements to be presented in terms of higher abstraction levels, intuitively closer to human-understandable concepts and heuristic decision-making. By attaching a set of features (also described in terms of taxonomy) to each measure, we show how this automated mapping can be optimized to provide the best available and suitable measure to a given measurement requirement. We illustrate this approach in terms of a security assurance example of Push E-mail service.

The rest of the paper is structured as follows. The following section describes some background concepts and related work. Section 3 describes our approach to Abstraction Layer. Section 4 presents a Push E-mail case example. Section 5 discusses findings of the study including their applicability, and finally, the conclusions summarize the paper.

2. BACKGROUND AND RELATED WORK

The BUGYO Beyond security assurance methodology defines a model for operational security assurance of services, motivated by well-known related standards, such as the Common Criteria (CC) ISO/IEC 15048 Standard [4]. While CC considers the design and implementation phase of a product or system, the objective of BUGYO Beyond methodology and tools is to evaluate if security control realization is running and deployed as expected in the operational system.

The security assurance model can be composed of several partial models or a single complete model. In any case, this model is defined at an abstract level that does not define the exact devices or some other elements deployed; it allows one to pick a chosen description level, such as service-level description. However, the abstraction needs to be mapped to the actual measurements provided to enable meaningful and adaptive operational management of evidence.

Utilizing relevant security, quality and security assurance taxonomies is core to our approach. Applicable taxonomies include especially security control taxonomies, security metrics taxonomies, security assurance taxonomies, information assurance taxonomies, and general Quality of Service (QoS) taxonomies and ontologies. Note that even though *security assurance metrics* are different from *security metrics*, security metrics can be used to offer evidence for security assurance metrics. For example, a security assurance metric offering evidence on the rigor of security assurance activities can use 'direct' security metrics to offer 'raw data' for the assessment of rigor. The QoS taxonomies concentrate on several quality attributes, such as performance, reliability, dependability, not only for security. Examples of several security ontologies are listed by Blanco et al. in [5]. Furthermore, Evesti et al. [6] compare few security ontologies from measuring and run-time applicability viewpoints. From these ontologies only work by Savolainen et al. [7] takes security metrics into account, presenting taxonomy for information security issues in service centric systems. The presented taxonomy categorizes security assets, attributes, threats, and solutions. Furthermore, the taxonomy contains a security metrics part that follows security metrics taxonomy proposal made by Vaughn et al. [8] for technical target of assessment metrics, yet splits metrics to strength and weakness metrics. However, the presented metrics classification is very general-level.

Similarly, security assets, attributes, threats, and solutions are listed also in Naval Research Laboratory (NRL) security ontology by Kim et al. [9] and in ontology of information security by Herzog et al. [10] The Ontology of Information Security – here called OIS – is the most extensive technically-oriented security control ontology that we found in the literature. The OIS contains 133 controls (or countermeasures), 79 assets, 88 threat and vulnerabilities, and 34 relationships between those concepts. Savola [11] proposes a holistic taxonomization model for security metrics in software-intensive systems. The purpose of this taxonomy is to model objectives of security metrics. The taxonomy presents objectives of security metrics in five levels as follows: (i) target under investigation, (ii) main viewpoint to target, (iii) fundamental measurement objectives, (iv) decomposition, and (v) more detailed metric characteristics. Level 3 – fundamental measurement objectives – is divided to three groups, i.e., security correctness, security effectiveness, and security efficiency.

GEMOM (Genetic Message-Oriented Secure Middleware) [12], [13] was an EU FP7 ICT project (2008-2010) that focused on significant increases in resilience, security, adaptation, intelligence and scalability within a Message Oriented Middleware (MOM). The core advances of the GEMOM approach are centered on adaptive security management technologies and security metrics. The GEMOM environment includes an operational Security and QoS Monitoring Tool (MT) [14], utilizing a collection of security and QoS metrics, defined in [15]. The MT implements a dynamical operational monitoring application. The MT interacts with a higher-level Adaptive Security Manager (ASM) node which can control several GEMOM brokers. Applicability of the developed security metrics is analyzed in a GEMOM banking money transfer scenario in [16]. An MT is connected directly to GEMOM Brokers. Connection between the MTs and Brokers is arranged via special client interface. Other entities use the GEMOM publish/subscribe mechanism to communicate: publishing and subscribing to relevant topics in a so-called measurement namespace. Although the GEMOM MT uses security metrics as a basis for ASM decisions, there are similar challenges in of utilizing metrics in this environment as in a security assurance MFW: the aggregation of sub-metrics is not straightforward, and the measurement requirements change dynamically.

3. PROPOSED ABSTRACTION LAYER

A core question in the deployment of the security assurance MFW is: *How can we best describe such measurement requirements and map them to available and attainable measurements from a deployed measurement infrastructure?*

Security assurance measurements often require aggregation of several metrics, because direct measurement of the relevant properties is not often possible in practical complex systems. Aggregation strategies can change from time to time, depending, e.g., on the predicted impacts of security risks (which, in turn, affect the security assurance strategy). Security assurance of a service is modeled using an AP, and the actual security assurance model of a system is composed from one or more such (partial)

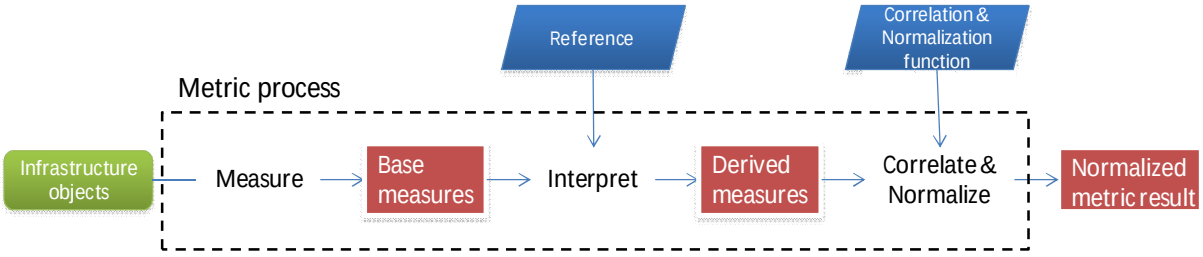

Figure 1. Base Measures and Derived Measures.

models. The measurement needs of this model have to be adapted to the measurement architecture of the target system. As the target systems change in time, the MFW must be capable of following such changes. An adaptive abstraction method is needed to fulfill the above mentioned needs.

An AP of the target system contains *measurement requirements*, based on the assurance needs that might change dynamically. Term 'measurement requirement' refers to the specification of measurements to be performed for each infrastructure object in order to achieve security assurance measurement objectives of the service. Moreover, there are a collection of *measurement probes*, which are available or not available in the MFW at different time instants. Probes deliver 0 to *N* basic measurements in a model-independent manner. This means that probes should have a standardized language, yet be abstract, not related to any specific model. During the operational security assurance evaluation, measurement requirements have to be matched dynamically with the available and attainable probes that can deliver the needed measurement results.

This abstraction mechanism can be implemented in the form of Abstraction Layer software element. We propose the following two-step Abstraction Layer approach: (i) Abstraction of measurement requirements: the measurement requirements are abstracted from measurements to allow for their linking to the ToM without having to specify the exact probe that is providing the measurement. These probes might change depending of the situation; (ii) Abstraction of measurement probes: the available probes are evaluated in order to enable selection of the probe best capable of fulfilling the measurement request.

3.1 Step 1: Abstraction of Measurement Requirements from Actual Measurements

We base our method of abstraction of measurement requirements from the actual measurements on a taxonomy describing the types of measurements that are identified as needed. Appropriate taxonomies discussed above help to maintain decomposition-composition relationships of security requirements, functions and controls, to bridge gaps between the highest abstraction levels and lowest-level measurable entities. Note that taxonomies are only as good as the quality of the *corpus* or source material [17] of them. Security concerns in different applications and services might require application of different taxonomies. Security risk is often very context-dependent and dynamical; therefore, the applicability of different parts of the taxonomies should be carefully analyzed during the operation of the system under investigation. Although security assurance relies on 'static' security objectives based on risk management decisions, it is important to integrate enough security risk awareness to the MFW to enable meaningful operational security assurance [18].

The measurement requirements can be implemented by one or more probes, utilizing Base Measures (BM) according to ISO/IEC 27004 [19] Standard terminology. The BMs are further aggregated to Derived Measures (DM) to provide a link to the AP, which defines measurement requirements in terms of composing BMs into DMs, and further into security assurance metrics for the service. Figure 1 visualizes the security assurance metric management process from infrastructure objects to metric results. A BM can be shared by several DMs and similarly, a DM can be shared by several assurance metrics.

The abstraction approach is illustrated in Figure 2, showing how DM is composed of several BMs. These are provided by an MFW, which translates the requests for a BM to actual probes it manages. Each of these probes is identified with BM identifier (*BM id*) that is composed of *Measure id* coming from BM taxonomy, and *Device id* that identifies which infrastructure object it is measuring. Note that the term *Device id* may be misleading at times as the target of measurement here can also be a service and as such hosted on or a subpart of the functionality hosted on a device.

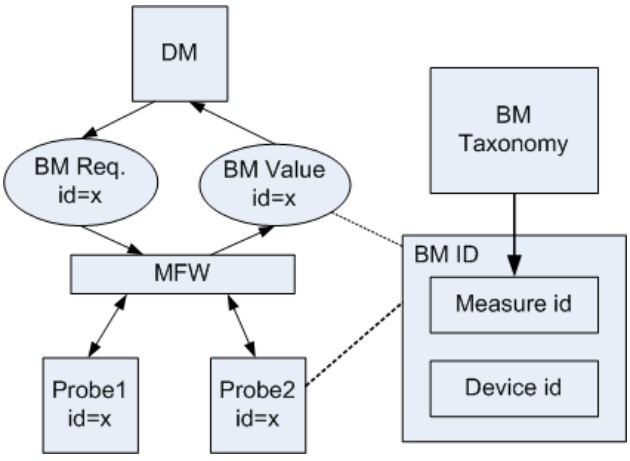

Figure 2. Taxonomy-based abstraction.

3.2 Step 2: Abstraction of Measurement Probes from Measurement Requirements

As a single measurement requirement at a high abstraction level may require one or more measurement results, we abstract the

measurement probes from the requirements by defining a number of measurement requirements for a single measurement objective.

As such, we may distinguish two categories of measurement probes depending on the relationship between the measurement tasks they purport to achieve: *collaborative* and *competitive* probes. The probes in the former category have to account for measurement values to meet a measurement objective. In this case, the results achieved by the probes are complementary in terms of deciding on an aggregated measurement objective value. The latter category is concerned with the selection of the more appropriate probe before the measurement can be performed.

In Figure 2, two probes, Probe1 and Probe2, are both assumed to be able to provide the required BM values. Additional information for the MFW to enable selection between one of the available probes (or BM) is given by meta-information what we refer to as *measurement characteristics information*. For example, a probe can be considered to be more suitable for fulfilling a given measurement requirement by checking its version against a database of available probe characteristics, or by its ability to cover more measurement requirements related to a measurement objective. Measurement characteristics information should include applicable information needed for security-relevant assurance evidence, such as relation to security control objectives, deployment characteristics of controls, and timing information.

In the following, we propose core structures for a standardized language can be used for the abstraction of probes. A probe can be associated with an eXtensible Markup Language (XML) file containing all relevant information, including measurement characteristics information. The XML file should contain the following sections:

- ⟨`probe`⟩ section,
- ⟨`measurement`⟩ section,
- ⟨`header`⟩ section,
- ⟨`description`⟩ section, and
- ⟨`measurement result`⟩ section.

⟨`probe`⟩ section will tell the essential characteristics of the probe itself (such as IP address, version numbers, and some other qualifiers like possibly load data in the probe's execution environment), while the ⟨`measurement`⟩ section contains information about what kind of measurements are delivered. The probe can have several ⟨`measurement`⟩ sections. Ideally, a ⟨`measurement`⟩ section contains information arranged using predefined object-oriented class templates, i.e., instances of a defined class.

⟨`header`⟩ section includes some structured meta-information of the delivered measurement, raising the need for a *naming scheme* capable of identifying the class of which the measurement is an instance of. ⟨`description`⟩ section incorporates an informal description of the characteristics of the measurement. Of course, if needed, the description can be informal or even formal, but for most practical cases, informal descriptions are required. This can even be free text written by the probe developer(s). A similar approach is applicable as used in Management Information Bases (MIBs) [20] of the Simple Network Management Protocol (SNMP), or Common Information Model (CIM) [21] of Web Based Enterprise Management (WBEM).

Finally, ⟨`measurement result`⟩ section will contain the measurement results in the desired form. Note that any section within ⟨`measurement`⟩ section can contain hyperlinks to other descriptions, and similarly, ⟨`measurement result`⟩ section can contain hyperlinks to other measurements on the same or other probes. In this manner, all measurement results have Unified Resource Identifiers (URI), or to be exact, Uniform Resource Locators (URL), for example, of type:

`bugyo://<probe_id>/measure/<measure-id`.

This URI is what will be used in the hyperlinks. Using this method, it is possible to make BMs dependent on other BMs, and of course, the agents delivering DMs can use this method to combine these values. Note that any such combination will be model-dependent, whereas BMs are not.

We do not define here in detail the 'body' information of the measurement results, because that information is dependent on the type of measurement and probe type. The 'body' can be, e.g., a table, a webpage, a report, an integer value, a binary value.

The above mentioned naming scheme could be based on diverse security assurance requirements in the Common Criteria, or other applicable standards, recommendations or taxonomies. Common Criteria approach is promising for naming scheme because it contains a class structure (e.g., functional requirements start with 'F' and assurance requirements with 'A'). The service-level security requirements are not mentioned in this scheme because this way the measurements would become model-dependent, causing problems when measurements are delivered to different measurement requirements.

4. Case Example: Firewall in Push E-mail Service System

In this section, we present an example describing how a security assurance measurement requirement associated with firewall functionality in a Push E-mail service system can be managed using the abstraction method proposed. We further discuss how the available measurements are described as a function of composition from several measurements, and how this composition can be used to choose between one of the several available measurements able to fulfill the measurement requirements and to describe to what extent the measurement requirements are fulfilled.

4.1 Push E-Mail

Our case example is a Push E-mail service system. The clear demand for 'always-on' capability in e-mail systems, especially in mobile phones and other hand-held devices, has resulted in Push E-mail systems [18]. In these systems, the emails for a user are 'pushed' to the mobile device from the email server as opposed to the mobile device always requiring querying ('pulling') the emails from the server. Description of the complete architecture of a Push E-mail system and all of its security assurance requirements is out of the scope of this paper; we focus on a single measurement point with specific measurement objectives in order to illustrate our approach. The approach needs to be scaled to include all the different elements across the network infrastructure, being essentially a case of aggregating all the measurements to the complete security assurance model.

We are also working on this especially in terms of the assurance profiles mentioned earlier and their composition into specific security assurance models for telecommunication systems.

A more detailed analysis of security metrics and security assurance metrics, which can be utilized in Push E-mail services in presented in [18].

4.2 Security Threats and Security Objectives

At the high level, the main security threats of Push E-mail service are (i) loss of service availability, (ii) Denial-of-Service (DoS) attack against the E-mail server, (iii) huge amount of spam e-mails, and (iv) disclosure of e-mail messages to unauthorized users.

Based on the identified security threats, the following high-level security objectives can be identified: (i) protect the integrity of E-mail Server, (ii) use appropriate spam filtering, and (iii) prevent unauthorized access to the client, e.g., a mobile phone, and the e-mail messages. Security requirements are constructed based on the security objectives.

4.3 Example of a Measurement Point

A measurement point is a point in the system under investigation, where one or more measurement probes are deployed. Figure 3 shows an example measurement point of security assurance monitoring system for our Push E-mail service infrastructure. In the example, it is assumed that there is a single measurement point: the firewall. Firewalls can be used to fulfill some of the security objectives discussed above. For simplicity, we do not discuss here the exact mapping from specific security objectives to the protective functionality offered by firewalls.

The system includes two servers: a web-server for the web-based e-mail interface, and the e-mail server itself. However, in practice, the measurement objectives associated with the firewall can be decomposed to three different measurement sub-objectives according to the system-level security functionality requirements: (i) objectives associated with the intranet, (ii) the Internet, and (iii) the Demilitarized Zone (DMZ). The AP includes measurement requirements associated to the measurement objectives. For simplicity, we do not address Domain Name System (DNS) here. For a Push E-mail example with DNS, see [18].

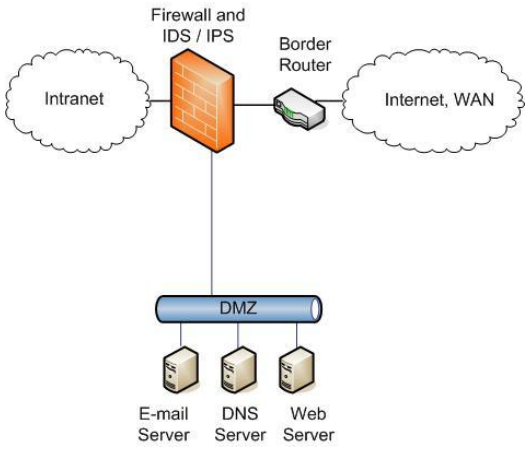

Figure 3. A Push E-mail firewall as a measurement point.

If a security requirement is defined using a general model like this, the actual implementation can follow different strategies. For example, one may deploy a single firewall to manage all relevant filtering and protection tasks identified in the objectives. On the other hand, one may deploy three different firewalls, each separate from one another. Furthermore, each firewall can be accessible from a single measurement point or may require several measurement points. The amount of deployed probes may also vary from time to time.

4.4 Abstracting the Measurement Requirements

Utilizing the security taxonomy described in [10], and specifically the classification of the security controls, i.e., countermeasures using the terminology in [10], to support measurement planning, we define the type of security control being observed as 'firewall'. Three measurement objectives can be identified for the TOM, termed 'intranet', 'Internet', and 'DMZ'. For the measurement objectives, we can then associate measurement requirements, for example, addressing the interfaces 'firewall, intranet', 'firewall, Internet', and 'firewall, DMZ'. Note that other approaches could also be taken to achieve higher granularity, such as defining firewall filtering policies as tuples of any of 'Internet', 'intranet', and 'DMZ'. For clarify and space reasons, we concentrate on the level of these three definitions of measurement points.

It must be noted that usually firewalls are part of the platform security than service security interest. For some service-related measurement requirements, a firewall could even be transparent. Moreover, the interfaces mentioned above are only examples of measurement points – after all, the whole service must be secured.

As we strive to provide measurements for a security assurance model element (firewall), we define the completeness of measurement as the ability of the MFW to provide measurements for the three measurement requirements discussed above. Examples of issues in operational security assurance of a firewall include: the basic rule set for filtering, usage of the firewall's internal resources, how does the CPU or memory load relate to the traffic it has, disk usage, internal logging, reboot activity, error situations, access control of it, the strength and management practices of administrative credentials.

Consider three different measurement points and the related requirements for their measurements. To illustrate the different concepts of our abstraction approach, we describe also two different scenarios of actual network deployment and its relation to the probes: (i) a single firewall managing all traffic, and (ii) three separate firewalls at each above mentioned measurement points. In the first case, we also consider that there is a single probe that is able to measure the whole configuration and other appropriate assurance-relevant information from the firewall. In the second case, three separate probes are needed to provide measurements for each of the three separate firewall configurations and other assurance information:

- In the single firewall case with a single probe that is able to perform the required measurements, we can arrive at an understanding that when the probe is available, the measurement requirements are fully fulfilled. Similarly, when the probe is not available, none of the measurement requirements are fulfilled.
- The three firewalls case with several probes is more complicated. If a probe that needs to provide results for one

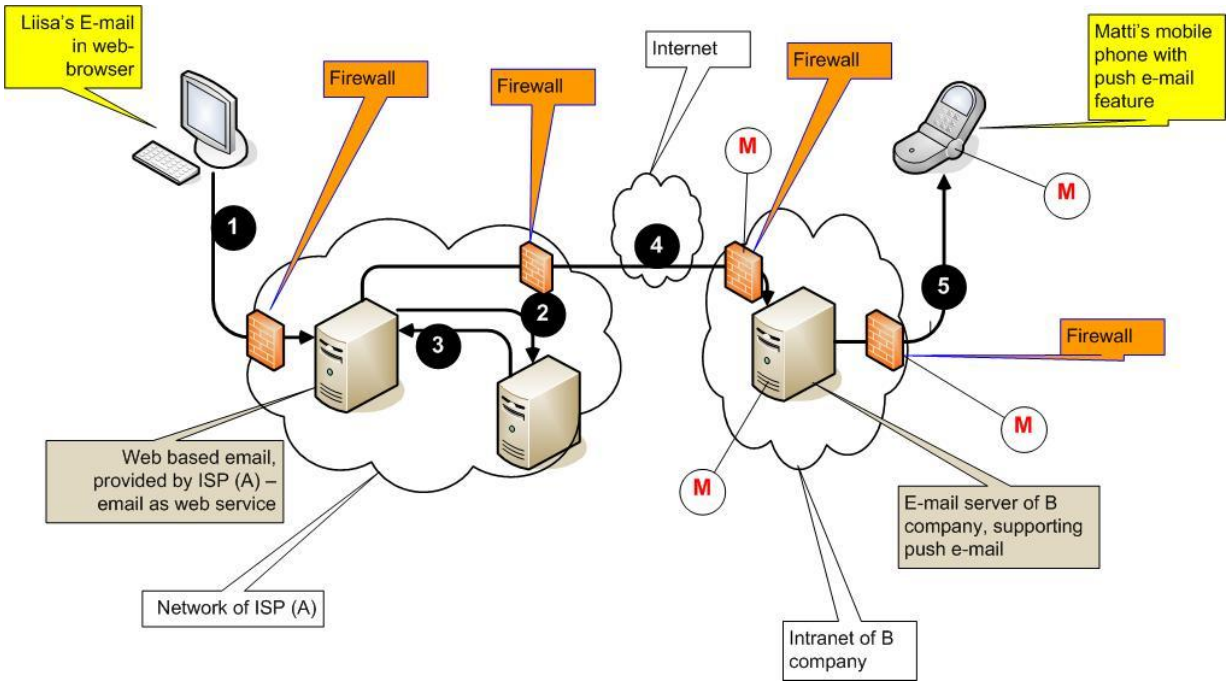

Figure 4. Example measurement points in Push E-mail Service system.

of the firewalls is not available, the related measurement requirement cannot be fulfilled. As there are three probes and three measurement requirements, this relates to how much of the measurement requirements can be fulfilled. Thinking simplistically, if only one can be fulfilled, we can say that the measurement requirements for the measurement point are fulfilled in completeness degree of 1/3. Similarly, for two or three functional probes the completeness reached is 2/3 and 3/3 respectively. In practice, the situation is more complex: what do you do if all three probes are available, but the data they deliver is only partly available? This could happen because semantically some measurements are not meaningful, have bias, or simply cannot be attained.

4.5 Use of Base Measures and Derived Measures

In order to illustrate the use of BMs and DM, let us investigate the element 'spam filter' of the Push E-mail system. For the purposes of checking the correct deployment of a spam filter, we can define two BMs: one for the investigation of correct deployment of the spam filter, and one for checking that the spam filter software is up-to-date. There are probes that can provide these BMs to the MFW, as shown in Figures 2 and 3. It should be noted that this kind of security assurance measurement is static. However, the Push E-mail service environment is dynamic, raising needs for adaptive solutions. There can be situations when more assurance information is needed about the functionality of spam filter, such as trend information of fault positives, depicting (partially) effectiveness of the spam filter. In this case, we add a probe for the measurement of the trend as a DM. If trend measurement is no longer needed, this probe can be removed from the use; otherwise, it may consume resources even though this measurement is not meaningful anymore. With the abstraction of measurement (see Figure 2), the dynamic requirements of the operational security assurance activities can be taken into account.

4.6 Combining the Measurements

Finally, we must be able to implement the measurement requirements of the complete security assurance model by aggregating all the measurements as they are specified here for the desired level of abstraction, e.g. security function. The overall architecture of our Push E-mail scenario is illustrated in Figure 4. This overall architecture includes a web-mail service where the e-mails can be accessed through a browser-based e-mail interface, as well as mobile devices where the emails are pushed to. In the figure, the measurement points of the Push E-mail system are labelled by red 'M's.

In the combination step, we need to aggregate the individual measurements from the overall system components, offered by the used measurement probes. Here the same considerations apply as at the lower level but with a higher abstraction level. Basically, we need to combine the DM values over the different components to produce the metrics describing the security assurance functions. This process is similar to that described before for the aggregation of BMs to DMs. The difference is that it requires aggregating the DM values with their completeness and other characteristics to form the metrics values that are used to provide the aggregated assessment of the security assurance level. However, the taxonomy of BMs is already aggregated at this level to DMs, and thus the definition of metrics at this level is more direct as the DM values are expected to be specifically matched to metrics without further abstraction (due to their specific nature).

It must be noted that aggregation of individual security-related metrics can be troublesome. Practical experience among industrial practitioners on aggregation has shown that the higher the

abstraction level, the 'greener' the results tend to be, assuming a three step traffic lights approach ('red' meaning low level of security assurance or level, 'yellow' mediocre, and 'green' good).

5. DISCUSSION

A key assumption of the approach presented here is that the changes perpetrated at the target system level can be accounted for by acting on the BMs at the measurement system level. In other words, adding, removing or updating a component for a system extension or for dwarfing a new threat could be handled by specifying, deleting or updating BMs on the monitoring system side. Although an automatic adaptation is highly desirable, in practice human intervention is hard to exclude at least when it comes to specifying or amending the BMs.

The described approach is defined for security assurance activities and the application is discussed in terms of this domain. Considering that there are many common elements to the more general security monitoring domain, a similar approach could be applied there by considering differently the measurement points and the completeness of measurement. For example, we could consider monitoring intrusions and other kinds of security attacks utilizing taxonomy of possible attacks, vulnerabilities and threats, associated with business or other relevant impact and risk exposure information. From this perspective, completeness of measurements could be considered in terms of how many identified attack vectors with remarkable impact or risk exposure are covered. Application over other domains could also be considered, but this activity depends on our ability to decompose measurement objectives in terms of taxonomy, ontology or a similar classification. For example, software and security testing compare the observed system behavior to the expected one, and composition of whole software systems from components can be considered. However, this would require highly domain-specific models and taxonomies, which limits the applicability of the approach in terms of high generalizations.

One aspect that would be worth investigating is the impact of the measurement framework on the performance of the evaluated system. The aim of developing taxonomy is to enable continuous and accurate security assurance measurements that reflect evolving system models. Nonetheless, a clear tradeoff analysis needs to be carried out between security, security assurance, and performance.

The applicability of the approach can also be analyzed by investigation of how well a mapping from the requirement model (the AP in our case) to a practically deployed large-scale system can be carried out. The same consideration also applies to the means to provide, and the availability, of such models for the system. An alternative approach can be simply to start from what is available as observed in terms of deployed measurement probes. We intend to further investigate these topics in the BUGYO Beyond CELTIC project.

6. CONCLUSIONS

We have proposed a conceptual approach to implement Abstraction Layer between the measurement requirements connected to security assurance activities, and the available and attainable measurements from the measurement infrastructure. Abstraction is needed to be able to adapt to different measurement needs and operational changes.

Taxonomies help to maintain decomposition-composition relationships of security requirements, functions and controls; and bridge gaps between the highest abstraction levels and lowest-level measurable entities. The proposed Abstraction Layer is based on abstraction of measurement requirements and probes from each other. In this context, we proposed and analyzed core structures for an XML-based standardized language for abstraction of probes, containing structures for probe identification, measurement and measurement results description, and applicable meta-information.

REFERENCES

[1] BUGYO Beyond CELTIC Eureka Project. Website available: www.celtic-initiative.org/Projects/BUGYO-BEYOND/default.asp [July 20, 2010].

[2] Zuccato, A., Marquet, B., Papillon, S., Aldén, M. 2006. "Service oriented modeling of communication infrastructure for assurance," Proc. of IEEE Workshop on Information Assurance, United States Military Academy, West Point, N.Y., USA.

[3] Marquet, B., Dubus, S., Blad, C. 2010. "Security assurance profile for large and heterogeneous telecom and IT infrastructures," Proc. of the 7th International Symposium on Risk Management and Cyber-Informatics (RMCI '10), Orlando, Florida, USA.

[4] ISO/IEC 15408-1:2005. 2005. "Common Criteria for information technology security evaluation – Part 1: Introduction and general model," ISO/IEC.

[5] Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Toval, A., and Piattini, M. 2008. "A systematic review and comparison of security ontologies," ARES '08, pp. 813–820.

[6] Evesti, A., Ovaska, E., and Savola, R. 2009. "From security modeling to run-time security monitoring," European Workshop on Security in Model Driven Architecture (SEC-MDA '09), CTIT Centre for Telematics and Information Technology, pp. 33–41.

[7] Savolainen, P., Niemelä, E., and Savola, R. 2007. "A taxonomy of information security for service centric systems," Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA '07, pp. 5–12.

[8] Vaughn, R.B., Henning, R., and Siraj, A. 2003. "Information assurance measures and metrics – state of practice and proposed taxonomy," Proc. of Hawaii International Conference on System Sciences.

[9] Kim, A., Luo, J., and Kang, M. 2005. "Security ontology for annotating resources," OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005 – On the Move to Meaningful Internet Systems 2005. (Agia Napa, 31 Oct. – 4 Nov. 2005), pp. 1483–1499.

[10] Herzog, A., Shahmehri, N., and Duma, C. 2007. "An ontology of information security," International Journal of Information Security and Privacy, Vol. 1, No. 4, pp. 1–23.

[11] Savola, R. 2009. "A security metrics taxonomization model for software-intensive systems," Journal of Information Processing Systems, 5(4), 197–206.

[12] Abie, H., Dattani, I., Novkovic, M., Bigham, J., Topham, S., and Savola, R. 2008. “GEMOM – Significant and measurable progress beyond the state of the art,” ICSNC ’08.

[13] Savola, R., Abie, H., Bigham, J., and Rotondi, D. 2010. “Innovations and advances in adaptive secure message oriented middleware – the GEMOM project,” RDCS ’10.

[14] Savola, R. and Heinonen, P. 2010. “Security-measurability-enhancing mechanisms for a distributed adaptive security monitoring system,” SECURWARE ’10.

[15] Savola, R. and Abie, H. 2010. “Development of measurable security for a distributed messaging system,” International Journal on Advances in Security, 2(4), 358–380, Publ. 2010.

[16] Blasi, L., Savola, R., Abie, H., and Rotondi, D. 2010. “Applicability of security metrics for adaptive security management in a universal banking hub system,” MeSSa ’10.

[17] Vogel, C. 1998. “Cognitive engineering,” Masson, Paris, France.

[18] Savola, R., Pentikäinen, H., Ouedraogo, M. 2010. “Towards security effectiveness measurement utilizing risk-based security assurance,” ISSA ’10.

[19] ISO/IEC 27004:2009. 2009. “Information Technology — Security Techniques — Information Security Management — Measurement,” ISO/IEC.

[20] Institute of Electrical and Electronics Engineers (IEEE). “SNMP MIBs,” Available: www.ieee802.org [July 20, 2010].

[21] Distributed Management Taskforce. 1999. “Common information model (CIM) specification,” Version 2.2.