

A MULTI-LAYER ARCHITECTURE FOR A SECURITY MANAGEMENT INFRASTRUCTURE

Michael Kretzschmar¹ and Frank Eyermann¹

Abstract. IT security is a matter of paramount importance especially to military organizations. Today's networks feature a large number of different security solutions—often not interoperable, complex to manage and laboriously to change or modify. This mostly leads to stovepipe systems with less flexibility and increased security concerns. Security Management Infrastructure (SMI) is an emerging research area that aims at assisting organizations in managing their security capabilities consistently and in provisioning security functions to organizational entities. In this paper we introduce a multi-layer architecture for an SMI integrating various inhomogeneous security devices and services of an organization and providing a uniform interface for accessing them. This new approach establishes the basis for a global and consistent management of the security infrastructure according to organizational goals.

1 INTRODUCTION

Security of IT assets is of ever increasing importance for most businesses as well as non-governmental and governmental organizations. Particularly the military needs to secure its IT infrastructure in order to protect data as well as to ensure the operational state.

Therefore, current network environments have to incorporate a growing variety of security services, devices, processes, and protocols. Especially in large military organizations, including joint and combined components, security measures were often implemented step-by-step or established only on demand, if mission requires, as “point-solutions” [1]. They solve one specific task, sometimes only for one dedicated user group, not looking how the chosen solution fits in the security architecture of the organization. The resulting environments are complex because the different security pieces from a diverse range of manufacturers are usually not compatible.

As a result many of these organizations operate on an inhomogeneous and non-interoperable security infrastructure, containing stovepipe systems, and application- and task-specific ‘security silos’ [2]. Applications using those security capabilities are implementing different standards or proprietary protocols and have to be changed if security components are replaced. Furthermore, each ‘silo’ or “stovepipe” must be configured individually—a laborious and error-prone task.

One solution to overcome this situation is an overarching security architecture integrating security capabilities and managing them according to an organizational security policy [3]. Such an architecture is also required to provide the flexibility necessary in future dynamic military operations [4]. This security architecture is termed *Security Management Infrastructure (SMI)* and is expected to solve the challenges [5].

Therefore, an SMI needs to provide two main functions: (1) an organization-wide security middleware incorporating the inhomogeneous security devices and services and providing a single, transparent and implementation-independent security functions interface for being used by enterprise applications, and (2) an also uniform management interface for this middleware and the incorporated security services and devices [1]. The uniform management interface allows configuring and managing information from these security

services and devices in an efficient and consistent manner and thus enhancing the organization's security.

Having these challenges in mind SMI can be defined as follows: A Security Management Infrastructure (SMI), also known as Enterprise Security Management (ESM) [6], is an infrastructure providing unified access to the security capabilities of an organization. It also comprises the systems and resources required to order, create, disseminate, modify, suspend, and terminate the management controls, which are necessary to provide and operate security services, devices, and processes across the organization according to a security guideline [5,7]. SMI is characterized by many heterogeneous, reused, and flexible security capabilities of different granularity and in different lifecycle phases within and across organizational boundaries. According to [5,7,8] SMI capabilities include, for example, Identity Management, Privilege Management, Metadata Management, Policy Management and Cryptographic Key Management. These security capabilities enable and manage basic security functions, such as perimeter defense, confidentiality, virus protection, protection of data at rest, or encapsulation of data during transmission [9].

In this paper we introduce a multi-layer architecture for an SMI system, which integrates the various inhomogeneous security capabilities of an organization and provides a uniform SMI API for usage by the organizational entities. Such an approach incorporating all concrete implementations of security capabilities does not yet exist. Current solutions cover only a limited scope and lack the necessary flexibility for future dynamic military operations. The SMI thus establishes the basis for a global and consistent management of all security assets according to organizational goals and requirements.

The paper is structured as follows: In Section 2, a scenario showing a joint and combined task force mission is presented. This scenario motivates the requirements for an integrated SMI and presents the respective management challenges. Following this, in Section 3 a requirements analysis is done, which guides the evaluation of related work in Section 4. Afterwards we describe the design of an architecture for an SMI system in Section 5. Finally in Section 6 we present our future research goals.

¹ Institut für Technische Informatik, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany

2 SCENARIO

An IT security infrastructure of a joint and combined task force for a stabilizing mission is described within this section. It should assist the local government of the host nation in establishing a secure and stable environment. The mission also includes external partners, which help accomplishing the mission goal (for example, armed forces or police of the host nation, or non-governmental organizations such as humanitarian groups, which permanently or temporarily support the mission).

Part of the task force is an allied headquarter, which operates a communication and information infrastructure. This IT infrastructure also includes various security devices, services, and processes to manage IT security—collectively termed as *security capabilities*. For the purpose of our work we define additionally the following terms: (1) a *security service* is a piece of software considered independently and which can be composed of various sub-services (for example, a Credential Management service may include a service to create X.509 based certificates), (2) a *security device* is hardware with software running on it (for example, a firewall or VPN gateway appliance) and (3) a *security process* builds upon functions of these services and devices and provides more complex functions (for example, an authorization process with policy checking) [5,10]. External partners are allowed to use specified resources of that infrastructure under the terms of a mission security policy. The security capabilities comprise multiple different security processes, services, and devices from various manufacturers that have been tailored to specific mission tasks. Each capability, in its own way, performs the security functions that it was designed for. However, taken as a whole, the security services and devices are usually not compatible and often do not ‘talk to each other’ [1]. Moreover various security operators, analysts, planners, and managers of security capabilities all maintain their own perspectives and issues [11].

The scenario is visualized in Figure 1. It shows the Communication and Information Infrastructure of the Headquarter, which is protected by several security services (Security Service 1 to n), security devices (Security Device 1 to n) and several security processes building upon the

services and devices. The security infrastructure also controls the access of external partners.

The interrelationships of the different security capabilities should be depicted by analysing a sample security process—an access request of a user in the role of a mission planner who wants to gather best practice solutions from archived information of previous military missions (see right half of the figure).

Before the mission planner is granted access to the secured database he needs to authenticate. In this example the user has a X.509 certificate including the appropriate private and public keys on a smartcard. The provided identity reference is double checked with a LDAP server (Light-weight Directory Access Protocol) in order to confirm whether the identity is still valid. Between the device of the user (for example, computer, PDA) and the authorization tool a SAML (Security Assertion Markup Language) assertion is created, which implements role-based Access Control policies defined in XACML (eXtensible Access Control Markup Language). Beside the information within the SAML assertion, the authorization tool requests additional metadata about the secured database and afterwards computes the corresponding access control policies for the access decision. Furthermore additional user and resource attributes are queried from a MySQL attribute management database. All this information is taken by the Policy Decision Point (PDP) to decide on the access request of the mission planner, which is then forwarded to the Policy Enforcement Point (PEP) implemented in the mission archive database.

Access requests to data by external partners must be handled differently, if they have not been issued a smartcard with a certificate of a trusted certification authority. Furthermore, not all applications might rely on an external PDP for an authorization decision but implement authorization as a core component. In this case they will also store the authorization profiles in their own proprietary form. If components of the security process and used protocols change (for example, the LDAP server is replaced by an Active Directory server) applications and configurations at various places in the network may need to be changed.

A consistent configuration of all security processes can be

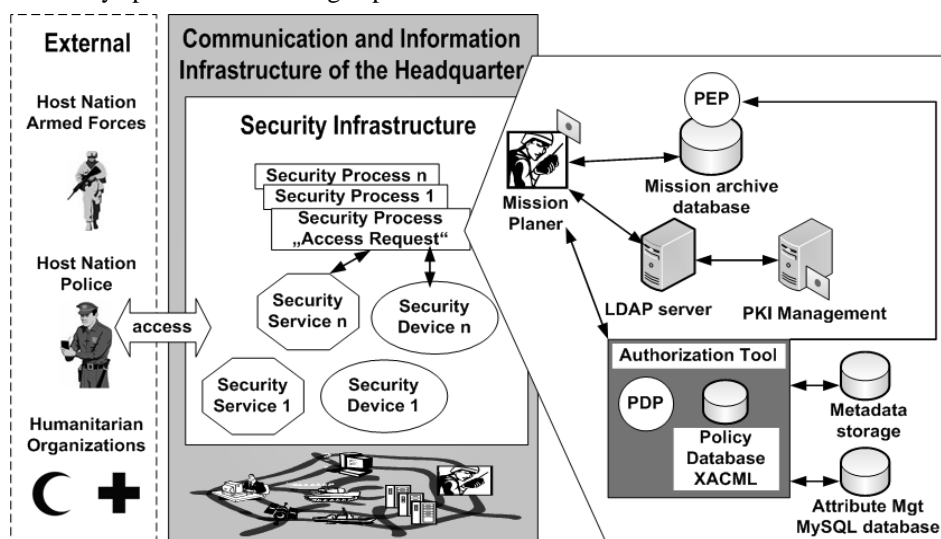


Figure 1. Security Infrastructure within a joint and combined task force.

accomplished today only in manpower-intensive manual configuration steps performed by various security operators, administrators, and analysts [5]. As military organizations of the future will rely on operational agility as a fundamental strategy in dealing with any type of adversary [4], systems and configurations may need to be adapted more frequently and on very short notice (for example, if a nation suddenly requests to leave the task force because of internal political affairs).

3 REQUIREMENTS ANALYSIS

This section defines the requirements of a unified and collaborative Security Management Infrastructure, based on but not limited to the scenario described in the previous section.

First of all, SMI needs to provide a single uniform interface to all security services and devices implemented in the organization. All persons and applications of the organization can make use of this interface, without knowledge of which concrete device and service are providing the information at the end. SMI should also be able to provide security functions for all kinds of internal or external entities including human or non-human users like standalone, two-tier, three-tier and Web applications.

Therefore, it is necessary that SMI is able to accommodate all types of security services and devices including Web- and not Web-based ones. This will foster the move from manual to automated, or when technically feasible and operationally viable, to automatic security management operations—also termed self-management capabilities [5].

As the security services and devices, on the one hand, and the number of potential users of an SMI, on the other hand, can get quite high in big or federated organizations, a scalable architecture is required. Moreover the number of security services and devices managed by one SMI must be changeable during the SMI's life time. Additional services and devices may need to be configured if the organization grows or if components are replaced. But also the set of supported devices and services must not be static. Continuously, new types of security services and devices evolve and manufacturers are extending their product portfolio. Furthermore, new kinds of applications may have additional security requirements, which are not yet taken care of. Consequently, an open, extensible, and flexible architecture is necessary, which can easily accommodate new types of services and devices as well as a changed set of the configured ones.

The security processes build upon the functionalities provided by the security services and devices. Their definition heavily depends on the security policy of the organizations (for example, authentication requirements, authorization rules, or encryption policies) but also on the mission needs (for example, providing access to data for first responders in case of a terror attack). Hence, if the security policy of the organization changes, SMI must support adapting the security processes easily and during run time.

As shown in the above scenario, additional complexity arises, if different organizations cooperate (for example, in a joint task force, or when using outsourced security capabilities). Therefore inter-SMI data exchange is necessary, where SMIs of two or more organizations share information (e.g., for

providing secure end-to-end communication). The use of standardized and non-proprietary protocols to communicate and exchange information between security capabilities will support this inter-organizational sharing of information.

The management of SMI has two aspects: (1) the operative management of all security capabilities, and (2) the management and configuration of the SMI system itself. SMI will allow a central management of all operative security data. Therefore the SMI API needs to provide an interface to all security-related data stored in the concrete security services and devices (for example, user accounts, passwords, or security policies). By concept of SMI, the interface must be independent of the services and devices used.

The SMI management must be able to monitor and change the behaviour of the SMI middleware, including the security processes and which concrete security services and devices to incorporate. This comprises the management and support of any security capabilities along their full life cycle—planning, provisioning, operation, modification and withdrawal.

4 RELATED WORK

The following section is structured into two parts. First we provide some theoretical foundations concerning security management models and general design principles. Secondly, we present an overview of current SMI approaches and exchange standards.

Security management is defined as one system management functional area of FCAPS within the OSI management architecture (ISO 10164) [3]. Here security management functions are described generically in order to be implemented by security management tools. The ISO 17799 Part 2 (2002) established the code-of-practice and the specifications of an Information Security Management System (ISMS) [8], which presents a methodology for providing and managing security services. It provides guidelines on how a management framework for enterprise security should be implemented. ISO/IEC 27001 (Nov. 2005) has been prepared to reemphasize the code-of practice of ISO 17799 with few amendments and additions of controls that will enhance and improve the ISMS further [9]. A framework for IT service management as unified basis for the development of further concepts for service management is proposed in [10]. This framework provides fundamental principles for IT security service management.

A Service Oriented Architecture (SOA) packages functionality of various applications as interoperable services. This allows different applications to exchange data with one another. [12] presents a Service Oriented Security Architecture (SOSA) as a collection of security services forming a security infrastructure used by Web service providers. In addition the Data Centric Security Model (DCSM) abstracts from security services (e.g., authentication and authorization) and their underlying mechanisms into interfaces directly supporting central data management policies [9].

Different activities in military, industry, and governmental organizations regarding unified interfaces to and management of security services, devices, and processes can be observed [1]. These current SMI tools focus on managing the entire security equipment of an organization. They are trying to solve the problem of heterogeneity on the management plane

by providing management tools to support security administrators, operators and planers. Such commercially available systems include Symantec ESM 6.5 [13], IBM's Tivoli Secure Way suite [14], ArcSight's ESM [15], CA's eTrust SCC [16], BMC Control-SA [17], e-Security's OeSP [18] and HP's OpenView [19].

Beside these solutions, there are some standards and approaches for specific security areas. In [20] a Web-based security management system providing security management services for small and medium businesses through Application Service Providers is introduced. For the exchange of authentication and authorization data, standards as OASIS SAML [21], specifications of Liberty Alliance, and the Web Services Federation Language [22] are implemented with their main focus on Web-based services [23]. Furthermore in the security area of crypto key management the Key Management Interoperability Protocol (KMIP) [24] can be used. Security Services Markup Language (S2ML) [25] is aimed at creating a common language for sharing security information about transactions and end users between businesses. Implementation-independent security services are provided by the Generic Security Services Application Program Interface (GSS-API). It defines an application programming interface for accessing security services [26]. The GSS-API, however, is limited to authentication in a client-server architecture. Other security services are not taken into account.

To summarize each mentioned approach cannot fulfill all requirements put forward for an SMI. The commercial systems assist security personnel greatly, still (1) lots of configuration work, especially when interconnecting security mechanisms and components, need to be done manually and (2) the range of security fields supported is often limited. None of these tools is flexible and holistic enough to ensure the required level of interoperability and flexibility in dynamic military missions.

Some of the standards shown might help implementing an SMI system particularly if exchanging information with other systems. They, however, focus only on single security fields, and their dissemination varies greatly.

5 DESIGN OF AN SMI ARCHITECTURE

In this section an architecture for an SMI is designed, which is shown in Figure 2 and will be described in more detail in subsequent sections. The architecture consists of three hierarchical layers. The lowest layer is the *Adapter Layer*, including the adapters and the concrete security services and devices. In the middle resides the *Generic Security Service Layer*, containing different *Generic Security Services*. The highest layer is the *Security Process Layer*, which builds upon the *Generic Security Service Layer*. The *Generic Security Service Layer* and the *Security Process Layer* provide functionality for users and managers. The list of functions they offer is subsequently termed *SMI API*.

5.1 Adapter Layer

The main challenge during this design process is the high number of different possible concrete security services and devices, which have to be accommodated and supported by the SMI. The security services and devices may differ in two dimensions:

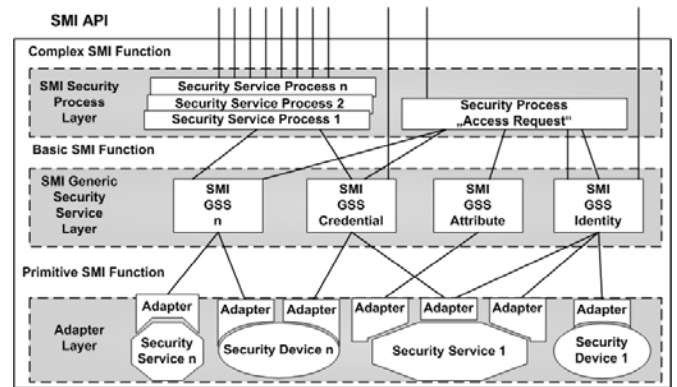


Figure 2. Multi-layer architecture for an SMI system.

- First, how a function of a service or device is interfaced with. Each service or device may have its own protocol for communication, for example, LDAP, proprietary APIs, or Simple Object Access Protocol (SOAP) in case of a Web service.
- Second, the function set that a concrete service or device provides. This set might be very different from service to service and from device to device. Furthermore the sets might mutually overlap.

The requirement to map all these different protocols and function sets to a uniform common SMI API bears therefore a high degree of complexity. The proposed solution is an abstraction and service decomposition layer, called *Adapter Layer*. It consists of different types of adapters, where each type may have a number of service- or device-dependent implementations. Each adapter has two interfaces: a primitive function interface, which is common for all adapters of one type, and a service- or device-dependent interface, which implements the (proprietary) interface of the concrete security service or device. A sample adapter type might comprise functions for changing firewall rules. Different adapters for, for example, Linux netfilter, Checkpoint Firewall 1, or Microsoft Internet Security and Acceleration Server have to be implemented.

For service decomposition it is required that the set of *Primitive SMI Functions* being implemented by the different types of adapters has to be disjunct. Furthermore, as a rule of thumb, the function set of one type of adapter should be chosen so that a concrete service or device either implements all functions of an adapter or none. Violation of this rule should be avoided but is not problematic. Each type of adapter can have multiple instances within the SMI system.

Figure 3 visualizes the decomposition of function sets in adapter types showing each primitive function as a symbol. In the upper part three services, e.g., providing identity management related functions, are shown. The service sets of the different services are not identical but they overlap – some functions are provided by all three services, some only by two, and some even only by one. As a first step during SMI implementation, adapter types need to be defined with disjunct function sets and with service-independent function definitions (middle part of the Figure 3).

In the next step service-dependent adapters need to be implemented (lower part of the Figure 3). In the case shown, a Service 1-dependent implementation of Adapter type A and B would be required and Adapter types B, C, and D need to

be implemented in accordance to Service 3. Service 2 is not shown due to space limitations. If available the adapter type will be defined and implemented according to standards like SAML or KMIP.

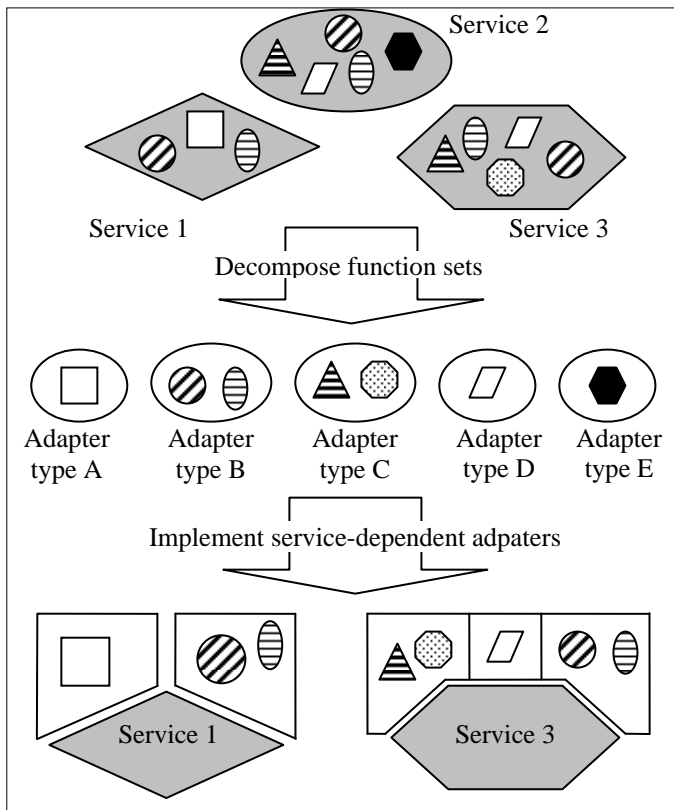


Figure 3. Decomposition of service sets in adapter types.

```

Adapter type D
interface identity_administration {
    void setIDName (in string IDName );
    string getIDName (in long IDReference);
    wstring getIDBiometricData (in long IDReference);
    boolean existIDName (in string IDName);
    boolean existIDReference (in long IDReference);
    ...;
}
    
```

Figure 4. Generic interface of a sample adapter type.

The reason for this service decomposition is reducing complexity in the layer above and when implementing adapters. High layers do not need to be aware of different function sets and different protocols – they only know a list of adapter types with a well-defined function list. Also adapter implementers profit from the decomposition as they can focus on implementing these function lists. In Figure 4 the stub of an interface of an adapter type providing identity management related functionality is visualized. A formal description approach (Interface Description Language, IDL) is used in order to maintain platform and programming language independence.

5.2 Generic Security Service Layer

The adapters map their set of *Primitive SMI Functions* to the next layer, the *Generic Security Service Layer* (GSSL). It comprises a list of *Generic Security Services* (GSS). A GSS represents one of the security fields such as Identity

Management, Cryptographic Key Management or Attribute Management.

Each GSS composes different *Primitive SMI Functions* to *Basic SMI Functions* and provides location transparency for SMI data. Location transparency means that the GSS implement algorithms to find transparently requested data among different concrete security services and devices, and therefore hides the distributed storage of data from higher layers.

An example for the location transparency can be visualized by referring back to Figure 2: Be Service 2 a Windows Active Directory server, which stores the user accounts for internal users such as the mission planner within the described scenario and Service 3 a SQL database storing the information about external user. For a concrete request the respective GSS needs to choose the right or all databases based on some criteria (e.g., username or function). If for example Adapter type D provides a search function, an appropriate GSS would need to query all services implementing an adapter of type D.

5.3 Security Process Layer

The *Security Process Layer* builds upon the *Basic SMI Functions*. It provides a framework for configuring complex security processes or security workflows easily and securely. This way security processes need to be implemented only once and not by each application using the process. The functions offered by security processes are termed *Complex SMI Function*.

Furthermore, they can be maintained centrally and adjusted quickly if security policies of the organization change. In contrast to GSS, which are by design independent of security policies of the organization, the security processes implement these policies and are subject to change. Changing the process, however, should not change the definition of the *Complex SMI Function*.

5.4 SMI API

The *SMI API* consists of the Basic and Complex SMI Functions. Both types of functions are exported by the SMI to users and administrators. Especially administrators will also make use of the *Basic SMI Functions*, as they require a more low-level access to the security data. An example of such a management function would be CreateIdentity, which creates a new identity reference in an appropriate security service. Based on parameters of a CreateIdentity request and the configuration of the SMI, the GSSL would forward this request to the right security service. In above example if creating an identity for an external user, the request would be forwarded to Service 3.

5.5 Evaluation of the Designed Architecture

The architecture presented in this section fulfils the requirements listed in Section 3. The *SMI API* provided by the GSSL and the *Security Process Layer* forms the single uniform interface to all security service, devices, and processes. The API can be used from internal human and non-human users, as well as internal and external ones.

The adapters of the *Adapter Layer* abstract from the concrete implementation of security devices and services. The

different adapter types decompose function sets allowing interfacing even when function sets differ greatly.

The layered approach also ensures the flexibility to exchange or add types of services, without change the *SMI API*. If new security functions are needed, however, new processes can be defined and the *SMI API* extended.

Management of the security services and devices is an integral part of the APIs within the SMI architecture as well as the *SMI API* provided. Therefore an SMI system should also be great help for security administrators, operators, and planners.

6 FUTURE WORK AND CONCLUSION

SMI is a challenging task, mainly because of the underlying infrastructure characterized by many heterogeneous, reused, and flexible security capabilities of different granularity and in different life cycles within and across organizational boundaries.

In this paper we introduce a multi-layer architecture for a SMI system. It integrates various inhomogeneous security devices and services of an organization and provides a uniform interface for accessing them. Thus it establishes the basis for a global and consistent management of the security infrastructure according to a common goal.

In our next steps we will focus on the description of the uniform interfaces of the three layers. For that we are analyzing various security environments to decompose functionality sets of security services and devices in order to describe a set of adapters, before the generic interface for each of the adapter is defined. Thereafter the resulting set of primitive functions provided by the adapter generic interfaces is clustered according to security areas to create Generic Security Services. Our future work will also include developing a prototype of a SMI system based on current security services, devices, processes and protocols running in the domain of the German Federal Armed Forces.

ACKNOWLEDGEMENT

We would like to acknowledge the support and contribution of the NATO SC/4 SMI AHWG. Furthermore this research activity has been performed partially in the framework of the EU IST Network of Excellence EMANICS "Management of Internet Technologies and Complex Services" (IST-NoE-026854).

REFERENCES

- [1] M. Nyanchama and P. Sop, "Enterprise Security Management: Managing Complexity", *Telecommunications and Network Security*, 2001, pp. 37-44.
- [2] G. v. d. Heidt and R. Schoepf, "PKI and Entitlement Key Information Security Management Solutions for Business and IT Compliance", *ISSE/SECURE 2007 Securing Electronic Business Processes*, 2007.
- [3] H.-G. Hegering, S. Abeck and B. Neumair, *Integrated Management of Network Systems – Concepts, Architectures and their Operational Application*, Morgan Kaufmann Publishers, ISBN 1-55860-571-1, 1 edition, 1999.
- [4] H. Janicke and L. Fich, "The Role of Dynamic Security Policy in Military Scenarios", *6th European Conference on Information Warfare and Security*, 2007.
- [5] NSA, *Enterprise Security Management: A Context Overview*, 2009.
- [6] B. Contos, "Deploy an enterprise-wide security solution with Enterprise Security Management (ESM) software", *Enemy at the Water Cooler*, Syngress Media, 2006.
- [7] NATO, *Concept for a NATO Security Management Infrastructure (SMI)*, document number AC/322-D(2008)0049 (INV), Dec 2008.
- [8] ISO/IEC, "Code of practice for information security management", *Internet standard, ISO/IEC 17799*, Dec 2000.
- [9] T. Grandison et. al., "Elevating the Discussion on Security Management", *Business-Driven IT Management BDIM '07*, 2007.
- [10] G. Dreo Rodosek, *A Framework for IT Service Management*, Ludwig-Maximilians-Universität München, Habilitation, Jun 2002.
- [11] D.A. Leaddston, "Enterprise Security Management – Reducing the Pain of Managing Multiple IDS systems", SANS Institute, 2004.
- [12] C. Opincaru, "Service Oriented Security architecture applied to Spatial Data Infrastructures", Universität der Bundeswehr München, Dissertation, 2008.
- [13] Symantec, Symantec Enterprise Security Manager, <http://www.symantec.com/business/>.
- [14] IBM, IBM Tivoli, <http://www-01.ibm.com/software/tivoli/>.
- [15] ArcSight, ArcSight Enterprise Security Manager, <http://www.arcsight.com/products/products-esm/>.
- [16] EMA, CA ESM, http://www.ca.com/files/IndustryAnalystReports/the_ema_all-stars_in_enterprise_systems.pdf.
- [17] BMC Software, BMC Control-SA, <http://www.bmc.com>.
- [18] e-Security Inc., Open e-Security Platform (OeSP) suite, <http://www.esecurityinc.com>.
- [19] HP, HP OpenView & Security Management, http://www.openview.hp.com/solutions/identity_mgt/sg/security_sg_jun03.pdf.
- [20] Y. Lim, M. Kim and A. Jeong, "An Enterprise Security Management System as an ASP Solution", *International Conference on Hybrid Information Technology*, 2006.
- [21] OASIS TC, SAML v2.0, sstc-saml-approved-errata-2.0, 2007.
- [22] BEA Systems et al, Web Services Federation Language (WS-Federation) Version 1.1, <http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf>, 2006.
- [23] W. Hommel and H. Reiser, „Federated Identity Management: Die Notwendigkeit zentraler Koordinationsdienste“, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bd. P-61, S. 6572, 2005.
- [24] OASIS, "Key Management Interoperability Protocol Usage Guide", Draft Version 0.98., <http://xml.coverpages.org/ni2009-02-27-a.html#kmip-spec-v098>, Feb 2009.
- [25] OASIS, "Security Services Markup Language (S2ML)", Technical report, <http://xml.coverpages.org/s2ml.html>.
- [26] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, Jan 2000.

Captain Michael Kretzschmar is an IT officer in the German Army and has graduated from the Universität der Bundeswehr München with a master's degree in Business Informatics. He is currently a PhD student focusing on federated identity management and information assurance for network-enabled capabilities. As corresponding author he can be contacted via email (michael.kretzschmar@unibw.de), phone (+49-89-6004-4764) or fax (+49-89-6004-3898).

Frank Eyermann has graduated from the Universität der Bundeswehr München with a master's degree in informatics and is currently a PhD student focusing on management of heterogeneous highly mobile ad-hoc networks in the context of software-defined radio.