

Article

Design and Implementation of a Hybrid Ontological-Relational Data Repository for SIEM Systems

Igor Kotenko *, Olga Polubelova, Andrey Chechulin and Igor Saenko

Laboratory of Computer Security Problems, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), 39, 14th Liniya, Saint-Petersburg, Russia;
E-Mails: ovp@comsec.spb.ru (O.P.); andreych@bk.ru (A.C.); ibsaen@comsec.spb.ru (I.S.)

* Author to whom correspondence should be addressed; E-Mail: ivkote@comsec.spb.ru;
Tel.: +7-812-328-2642; Fax: +7-812-328-4450.

Received: 26 April 2013; in revised form: 10 June 2013 / Accepted: 17 June 2013 /

Published: 9 July 2013

Abstract: The technology of Security Information and Event Management (SIEM) becomes one of the most important research applications in the area of computer network security. The overall functionality of SIEM systems depends largely on the quality of solutions implemented at the data storage level, which is purposed for the representation of heterogeneous security events, their storage in the data repository, and the extraction of relevant data for analytical modules of SIEM systems. The paper discusses the key issues of design and implementation of a hybrid SIEM data repository, which combines relational and ontological data representations. Based on the analysis of existing SIEM systems and standards, the ontological approach is chosen as a core component of the repository, and an example of the ontological data model for vulnerabilities representation is outlined. The hybrid architecture of the repository is proposed for implementation in SIEM systems. Since the most of works on the repositories of SIEM systems is based on the relational data model, the paper focuses mainly on the ontological part of the hybrid approach. To test the repository we used the data model intended for attack modeling and security evaluation, which includes both ontological and relational dimensions.

Keywords: ontology; security information and event management; data model; data representation; logical inference; repository

1. Introduction

The technology of Security Information and Events Management (SIEM) becomes, at present, one of the most important research directions in the area of computer network security. The essence of this technology is to provide an ordered collection of security log records from a variety of sources, their long- and short-term storage in a centralized data repository in a common format for modeling and analysis to detect and predict attacks, and developing countermeasures. Data analysis in SIEM systems is based, as rule, on the methods of event correlation, data mining, logical reasoning, and data visualization.

The usage of SIEM systems is very important for a large variety of information infrastructures such as distributed networks of Internet enabled objects (Internet of things). The examples of such networks, which were analyzed in our work, are the distributed computer network of transnational corporation, the infrastructure of mobile money transfer service and critical infrastructures, such as dams and power plants, where embedded devices and sensors are connected with data centers and servers by the Internet or mobile networks [1].

Existing well-known SIEM systems have multiple constraints of their usage in above networks and infrastructures. The most significant limitations are low scalability, restrictions on functions imposed the trust infrastructure, the inability of agreed interpretation of incidents and events at various levels, and the impossibility to provide high reliability and fault tolerance in distributed environments to capture event data.

To avoid the above shortcomings and ensure the effective application of SIEM technology, the development of advanced SIEM systems is required. In other words, we need to develop new generation SIEM systems, which remove functional infrastructure restrictions and have the ability to correlate events in cross-domain manner, high reliability, and durability of event data, as well as high scalability. In addition, researchers consider that one of the most significant functions of the new generation SIEM systems is the modeling of security events, attacks, and countermeasures [2].

The data repository is one of the main components of new generation SEIM systems, which is purposed to represent heterogeneous security events in uniform internal format, their storage in accordance with the previously developed data model, and supporting the extraction of relevant data for SIEM analytical modules. The overall functionality of SIEM systems depends largely on the quality of the solutions adopted in the repository implementation. An ability of new generation SIEM systems to meet these requirements when they operate in distributed networks depends on the approach and the data model that are used for design and implementation of the data repository.

In our opinion, the ontological approach to data modeling in addition to the relational approach is one of the most appropriate for this goal. It can be used to represent the SIEM data with complicated relational representation. Such data can be used to analyze the current security situation, model attacks, and generate countermeasures, including the analysis of historical data. This paper, as an example, considers the ontology of vulnerabilities, which can be used in analytical SIEM modules.

The ontological approach involves the use of a formalized description of the subject area based on description logics, known as *ontology* [3]. Ontological approach to data modeling is now increasingly used in many technical areas, including information security [4,5]. We assume that the hybrid approach, based on a combination of ontological and relational models, is promising for new generation SIEM systems, taking into account the fact that, currently, there are sufficiently mature

support tools for ontological data models. The confirmation of fairness of this idea, the analysis of possible solutions, and the consideration of the key issues for implementation of the hybrid data repository, which includes ontological components, is the main goals of the paper.

The rest of the paper is organized as follows. Section 2 presents the review of related work in the field of applying the ontology data models for information security. Section 3 analyzes existing SIEM systems and standards. In Section 4 we select the ontological approach as a core component of the repository. Section 5 describes our proposals for ontological data model of SIEM systems. In Section 6 and Section 7 we discuss the architecture and implementation of the hybrid repository respectively. Section 8 outlines an example of testing the repository for the task of attack modeling and security evaluation. Section 9 concludes our results.

2. Related Work

The analysis of related works on applying ontologies for network security helped us to select the following basic directions that can be used for SIEM systems: verification of security policies, intrusion detection, vulnerability analysis, security monitoring, and forensics.

The greatest popularity is the use of ontologies for verification of security policies. López de Vergara *et al.* [5] propose to use an ontology based on Intrusion Detection Message Exchange Format (IDMEF) to represent and share the knowledge about incidents. This ontology is considered as the first phase to develop the knowledge base, which should include policies, incidents, authorizations, *etc.*

Cruz *et al.* [6] propose an ontology to model the dynamic aspects of role-based access control. The information infrastructure of the Olympic Games is used as the scenario to assess the proposed approach. This use case is well suited to test SIEM capabilities.

Kolovski *et al.* [7] and Rochaeli *et al.* [8] suggest an ontological approach to engineering the security policies, using the “services-actors-resources” model and the paradigm of ontological templates. These papers show the preference of the ontological approach in comparison with other ones, in particular, with an approach based on propositional logic.

Fitzgerald *et al.* [9] outline the ontological approach for configuring the firewall management policy for Linux Netfilter. The results demonstrate that this approach is a reliable, convenient, and automated.

Taylor *et al.* [10] apply the ontological approach to represent complex events, including security events, in heterogeneous computer networks. This paper, taking the advantage of the logical reasoning based on description logic, proposes a solution to translate the formal specification of events using the native language.

Razzaq *et al.* [11] consider the possibility to use ontologies for intrusion detection. They show how the ontological approach allows detecting zero day attacks. The ontology base helps in focusing on a specific portion of network packets where an attack is possible.

Rochaeli *et al.* [12] propose an ontological approach to construct the knowledge representation system for computers and their vulnerabilities. Concepts and roles are described to represent the dependencies between the computer model and the communication mechanisms that have known vulnerabilities.

Schatz *et al.* [13] show that ontology is an effective tool for domain-specific, event-based knowledge specification in the automated forensics area. This is the case of cross-domain correlation.

The proposed approach integrates the ontology of standardized components that can simulate particular domains. The approach is applied to the scenarios including the enterprise resource transactions and computer security events.

Kenaza *et al.* [14] suggest the use of an ontology to provide contextual security event monitoring and intrusion detection. The ontology is used on data preprocessing phase to convert a set of warnings into a set of formatted data.

The positive results from these papers certify our ideas about implementation of the ontological approach in the repository of next-generation SIEM systems.

A distinctive feature of our idea is that we do not exclude the application of the relational approach in the repository, particularly for the data processing tasks that require real time. At the same time we believe the work with the ontological data representation can be more efficient than with the relational one in cases when we can realize logical reasoning in the repository.

Furthermore, the main interest in this paper is the ontological representation of data about vulnerabilities. It is envisaged that these vulnerabilities are loaded into a SIEM system from external databases and are used for attack modeling and security evaluation. The paper demonstrates an example of improving the performance of the SIEM system component, which uses a mixed data schema, where vulnerabilities are represented in the form of ontology.

3. Security Information and Events Management (SIEM) Systems and Data Formats

We investigated the most interesting solutions in data representation and storage in existing SIEM systems. For these goals, we have chosen OSSIM AlienVault, Cisco AccelOps, QRadar, Prelude, ArcSight Logger, IBM Tivoli SIEM, and Novell Sentinel Log Manager. These systems are on the top in [15]. The results of this analysis lead to the following conclusions.

First, all developers of SIEM systems listed above use relational databases. They implement such database management systems (DBMSs) as MySQL, PostgreSQL, and SQLite. Some of them declare that they can use XML databases.

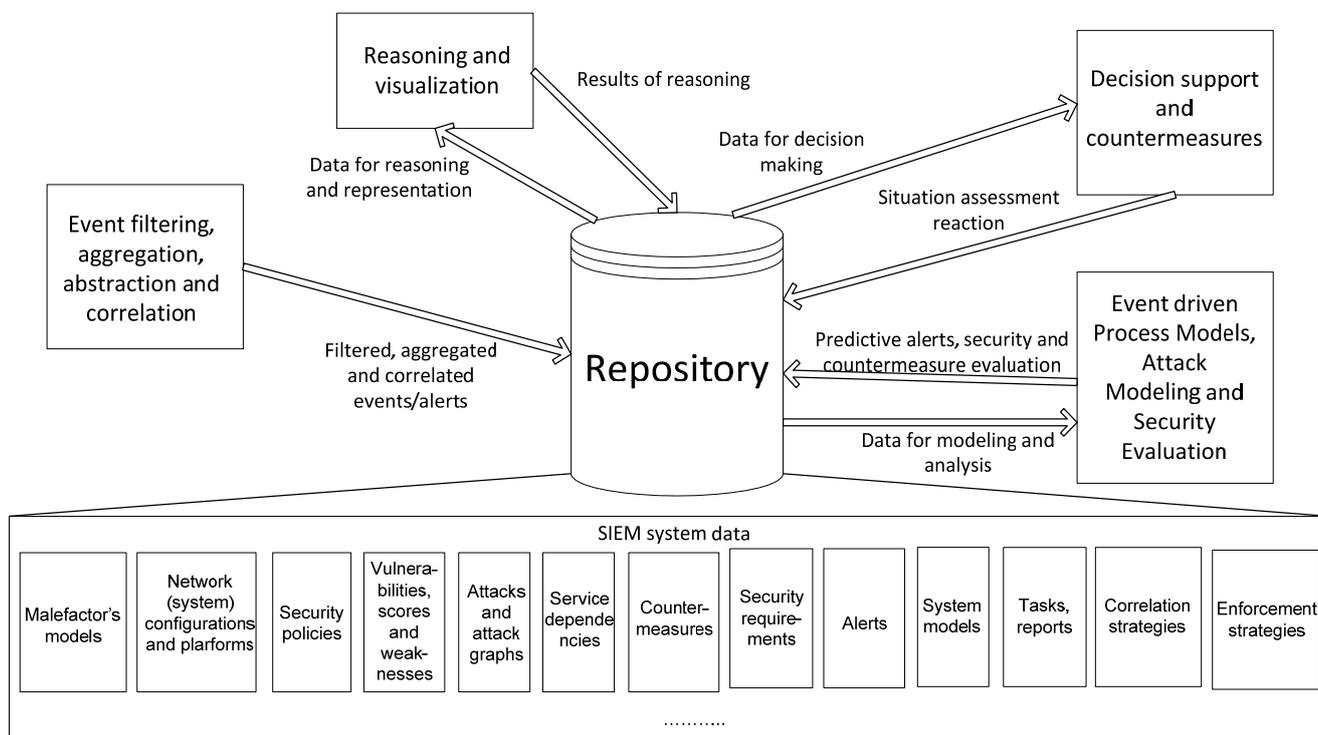
Analysis of advanced SEIM systems allows us to identify the location of the repository in the advanced SEIM system, and to identify the main flows of data between the repository and the rest SIEM components. These data flows are depicted in Figure 1.

Then, the analysis of advanced SEIM systems let us to conclude that the main SIEM components (Figure 1), which are sources and consumers of data stored in the repository, are:

- Event filtering, aggregation, abstraction, and correlation;
- Reasoning and visualization;
- Decision support reaction and countermeasures;
- Attack modeling and security evaluation.

Information and event management standards provide the most common rules to represent the security events and incidents. So as to be more comprehensive we investigated the following standards: Common Base Event [16], Common Event Format [17], Intrusion Detection Message Exchange Format [18], Common Information Model [19], and Security Content Automation Protocol [20].

Figure 1. Main data flows with repository in Security Information and Events Management (SIEM) system.



The Common Base Event model (CBE) is a standard that defines XML event syntax. CBE is a common language to detect, log and resolve system problems. It is supported by several Tivoli products. Since the public release of the specification and IBM’s partnering with Cisco in 2003, CBE has continued to be actively maintained.

Common Event Format (CEF) was proposed by ArcSight. It is an open standard that contains the most relevant event information, making it easy for event consumers to parse and use them. This format integrates various approaches to represent popular logs for infrastructure providers. The format is mostly oriented to ArcSight SIEM systems.

Intrusion Detection Message Exchange Format (IDMEF) is an Internet Engineering Task Force (IETF) effort. IDMEF was designed to enable the communication of intrusion events. It consists of two entities: the syntax expressed in XML and the transport protocol (Intrusion Detection Exchange Protocol, IDXP). First was proposed in 2002, with the most recent update occurring in 2004. IDMEF is supported by a limited number of intrusion detection products.

Common Information Model (CIM) provides a common definition of management information for systems, networks, applications and services, and is used for vendor extensions. CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network. CIM covers the widest possible scope among the listed standards. At the moment it is actively growing.

Security Content Automation Protocol (SCAP) [20] enables to compile a list of system platforms and applications, set their configuration, specify the list of vulnerabilities to assess the adverse effects of configurations and security vulnerabilities, identify the most critical vulnerabilities. It consists of a number of standards that describe:

- Features of software and hardware configuration (Common Platform Enumeration, CPE);
- Software and hardware configuration which adversely affects the security (Common Configuration Enumeration, CCE);
- Vulnerabilities of these products (Common Vulnerabilities and Exposures, CVE);
- Effects of configurations and security vulnerabilities, the most critical vulnerabilities, and corrections of errors (Common Vulnerabilities Scoring System, CVSS).

Under construction of the ontology described in the paper we have focused mainly on the SCAP protocol, as it includes a standard for describing vulnerabilities.

4. Ontological Approach

The paper focuses on the ontological approach (used in conjunction with the relational approach) and outlines its benefits to represent vulnerabilities.

In all known SIEM systems the relational approach is used for data storage. However, the relational approach has various constraints on expression of relations between entities.

As a rule, the relational model is often overloaded. This can lead to the conclusion that querying the data can take a long time. This is due to the lack of flexibility and expressiveness of SQL query language used in relational databases.

The next challenge of relational data modeling is the need to update the data schema, when the big changes of the subject area occur. For relational database systems, storing large amounts of data, this task requires a lot of overhead.

An example of vulnerability description represented by CVE is shown in Figure 2.

It means that the vulnerability occurs when the host has application “microsoft ie” and one of the following operation systems: “microsoft vista sp2” or “microsoft vista sp2 ×64” or “microsoft server 2008 sp2 ×86” or “microsoft server 2008 sp2 ×64” or “microsoft 7 ×86” or “microsoft 7 sp1 ×86” or “microsoft 7 ×64” or “microsoft 7 sp1 ×64” or “microsoft server 2008 r2 ×64”.

In the relational data model, the entire product list, describing the vulnerability, along with logical operators is stored as a row in a table (Figure 3).

This representation form does not allow flexibly specifying a parameterized query (with the names of products, versions, *etc.*) for the analysis of vulnerabilities.

An alternative solution on data representation in information processing systems with complex data structures (such as SIEM systems) is an ontological approach, which makes much easier the expressions of complex relationships between entities. When using this approach, the concepts and relationships can be formulated on the base of description logics. At that the terms of the dictionary are the names of unary and binary predicates (concepts and relations). Such predicates are used to express the relations between concepts and to limit their interpretation.

The *ontology* is a knowledge base that describes the facts that are always true as part of a community and are based on the generally accepted meaning from the dictionary. In the simplest case, the ontology describes only a hierarchy of concepts and relationships. The changes in the ontological data model require much less effort than in the relational model, especially in areas where it is needed to store different types of information that can be quickly changed. These areas, of course, embrace the security issues of distributed networks, including SIEM.

Figure 2. Example of vulnerability description represented by Common Vulnerabilities and Exposures (CVE).

```
<vuln:vulnerable-configuration id="http://nvd.nist.gov/">
<cpe-lang:logical-test negate="false" operator="AND">
<cpe-lang:logical-test negate="false" operator="OR">
<cpe-lang:fact-ref name="cpe:/a:microsoft:ie:9" />
</cpe-lang:logical-test>
<cpe-lang:logical-test negate="false" operator="OR">
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_vista::sp2" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_vista::sp2:x64" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008::sp2:x86" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008::sp2:x64" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::x86" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::sp1:x86" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::x64" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::sp1:x64" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008:r2::x64" />
<cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008:r2:sp1:x64" />
</cpe-lang:logical-test>
</cpe-lang:logical-test>
</vuln:vulnerable-configuration>
```

Figure 3. Example of vulnerability representation as a row in a table.

16	OR(cpe:/o:hp:Apollo_domain_os:sr10.2, cpe:/o:hp:Apollo_domain_os:sr10.3:beta)
17	OR(cpe:/o:sun:sunos:4.1, cpe:/o:sun:sunos:4.1.1)
18	OR(cpe:/o:sun:sunos:4.0.3, cpe:/o:sun:sunos:4.1, cpe:/o:sun:sunos:4.1.1)
19	OR(cpe:/o:sun:sunos:4.0.3, cpe:/o:sun:sunos:4.0.3c)
20	OR(cpe:/o:digital:ultrix:4.0, cpe:/o:digital:ultrix:4.1)
21	OR(cpe:/a:next:next:2.1)
22	OR(cpe:/o:attr:svr4:4.0)
23	OR(cpe:/o:digital:ultrix:4.2)
24	OR(cpe:/a:ncsa:telnet)

It should be also noted that when designing SIEM systems, the data model must be the most common and at the same time not overloaded one, which will be adapted and specified for each application area. Therefore, the use of ontologies is a necessary approach that enables to create a general model that can be flexibly and quickly applied for all necessary concepts in new generation SIEM systems. Loose coupling of domain ontologies makes it easy to add, delete and support individual ontologies. In addition, the components of the ontologies may be dynamically combined during the performance of SIEM systems to meet specific application requirements.

Mathematics underlying the ontological approach allows building more accurate queries and thus reducing the time spent by the analytical modules of SIEM systems to select information from the repository for a subsequent analysis. This advantage is particularly important in the field of distributed network security, because here there is a need to carry out in-depth analysis of heterogeneous information including historical records.

5. Data Model

As an example of an SIEM system component, which can use the ontological data model, the Attack Modeling and Security Evaluation Component (AMSEC) was chosen [21]. AMSEC generates

graphs of attacks using the models of the network and malefactors and the list of known vulnerabilities. Further, on the basis of the constructed graphs, the AMSEC evaluates the common security level for the network, and for each host of the network identifies weaknesses and assesses possible countermeasures aimed at increasing the security level of the network. The AMSEC also allows calculating the likely characteristics of the malefactor, predicting possible attack routes, and reconstructing malefactor actions performed before the current action. The SCAP protocol has chosen as the base for the AMSEC data model. On this basis we have built a relational and ontological parts of data model.

Let us describe the ontology of security vulnerabilities, which has allowed to speed up the retrieval of data from the repository, and to increase the performance of the AMSEC.

The fragment of the ontology, describing concepts, vulnerabilities, attacks, software/hardware manufacturers, and other concepts, is shown in Figure 4. It includes such concepts as *Vulnerability*, *IT_Vendor*, *Product*, *Reference*, *Basic metrics*, *CWE*, *CVSS*, and others.

The concept *IT_Vendor* is designed to represent software and hardware manufacturers. Based on the concept *Product*, a hierarchy of software and hardware, that can include vulnerabilities, is constructed.

The concept *Vulnerability* is for the specification of vulnerabilities.

The concept *Reference* is designed to link the vulnerability to its description in various public vulnerability databases.

The concept *Vulnerability_Product* is used to reason about the inclusion of vulnerabilities in software and hardware. A subclass of this concept is created for each vulnerability. It establishes the necessary links between subclasses of concepts *Product* and *Vendor*.

For example, *VP_1* describes a vulnerability CVE-2003-0497 (subclass of *Vulnerability*), which is derived for the software *Cache_database* (subclass of *Product*), developed by Intersystem (subclass of *IT_Vendor*). Thus, the property *hasEdition* of the concept *Cache_database* must be equal to five. More complex vulnerabilities (such as *CVE-2009-3115*), presented in Figure 4, are specified by the relationships of more concepts *Product* and *IT_Vendor*.

An example of the basic vulnerability affecting only one software product (Cache database version 5 of the Intersystem Company) is shown in Figure 5. This is the relationship between the concept *CVE-2003-0497* (from the CVE database) and the concept *VP_1*.

Developed ontology consists of a data schema called TBox (Terminology box) and the data themselves—ABox (Assertion box). The description of a vulnerability is a sequence of program-hardware components connected by Boolean operators (AND, OR, NOT, AND, NOT, OR). As it is shown in this figure, in the presented model the relationships between software and hardware components, which generate the vulnerability, are specified using the description logic.

Connections between the concepts are represented, mainly, by the subclasses (depicted as “is-a”), rather than through the properties of the objects. Thus, the logical reference here is confined to the task of classification, which increases the processing speed.

Figure 4. Ontological model for vulnerability representation.

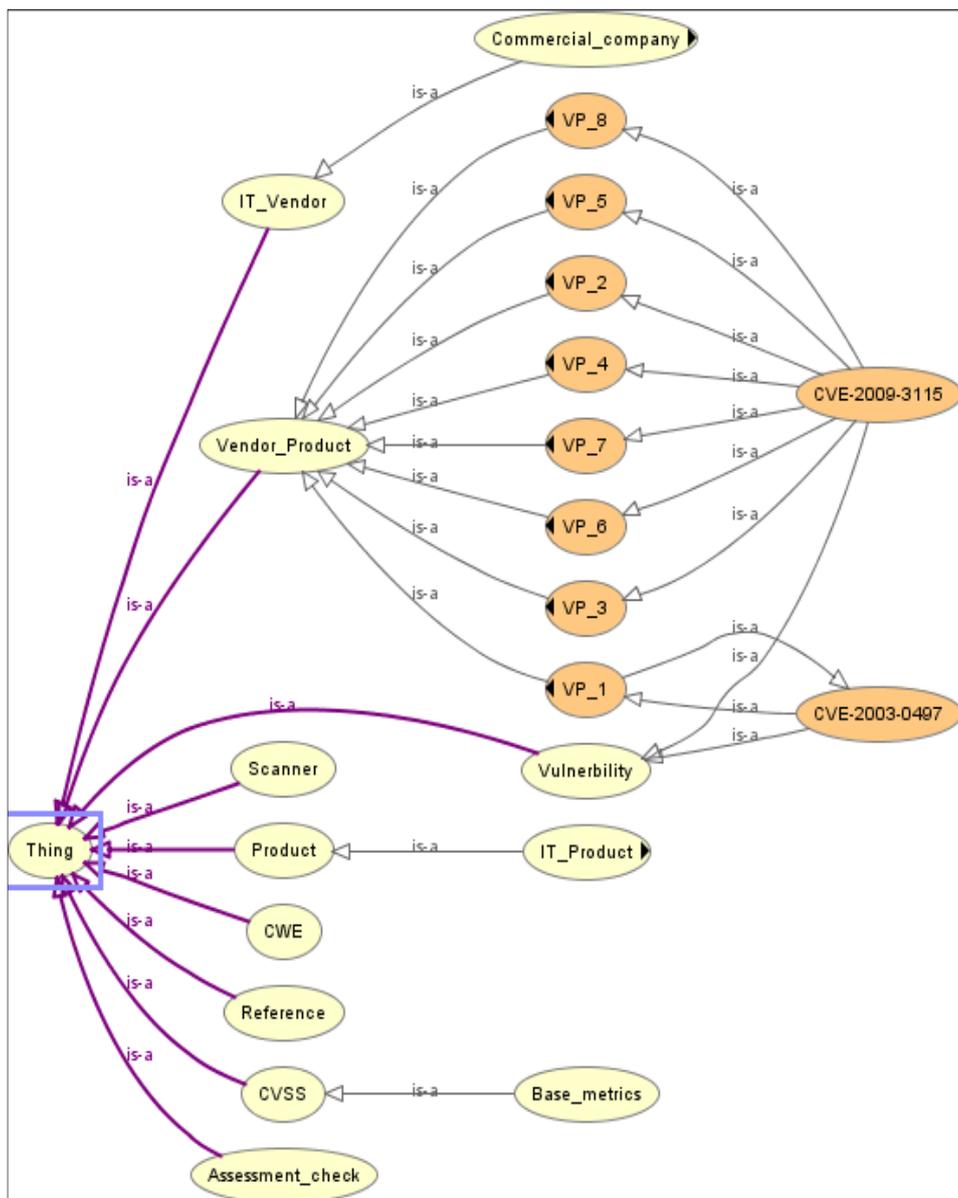


Figure 5. An example of ontological representation of vulnerabilities.



In our work we sought to reduce the expressiveness of the description logic to reduce the complexity of reasoning over the knowledge base, while expressing all the necessary properties and relationships. The expressivity of the description logic corresponds to the *ALEQ(D)* language.

This language includes the following features:

AL is an *Attributive Language*, a description logic that provides atomic concept, universal concept (*T*), bottom concept (\perp), atomic negation, intersection (\sqcap), value restriction ($\forall R.C$), and limited exist restriction ($\exists R.T$);

E—full extension quantification (existential restriction that has fillers other than owl:Thing);

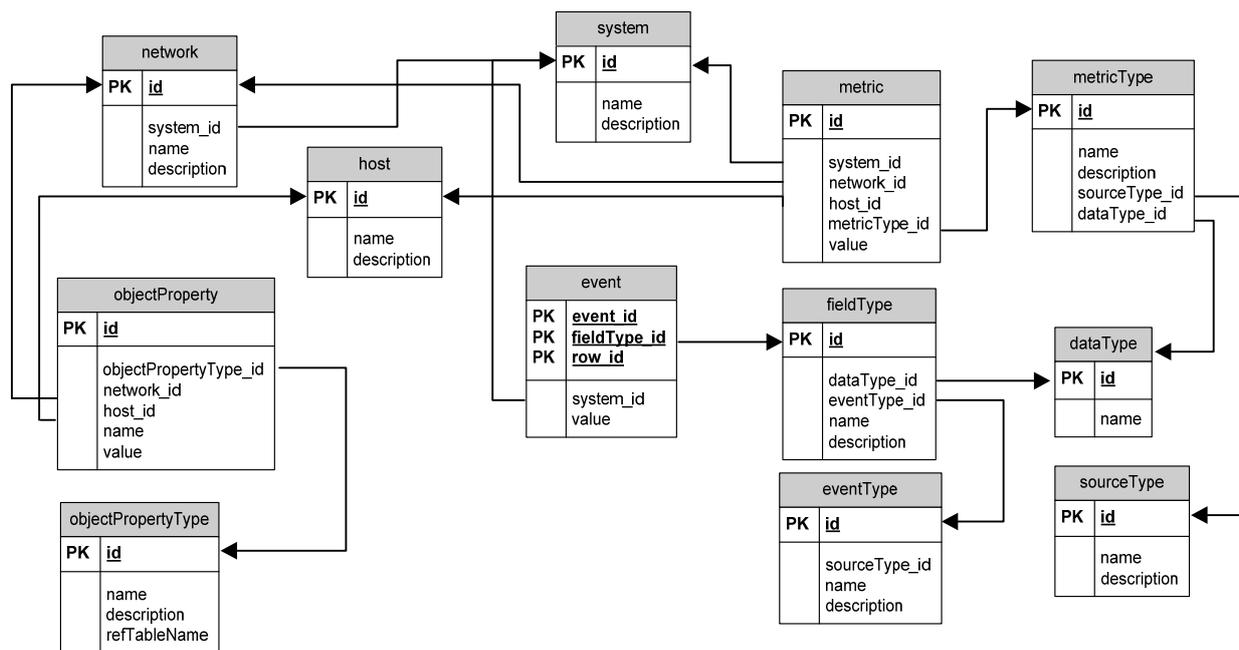
Q—qualified cardinality restriction;

D—use of data type properties, data values, or data types.

This description logic, among other characteristics, can use cardinality restriction and qualified cardinality restriction (available in OWL 1.1).

Next, let us consider the relational part of the overall scheme of the AMSEC, stored in the hybrid repository. It is represented in Figure 6. The data scheme, which is used for the AMSEC, contains a set of the tables: *network*, *host*, *objectProperty*, *objectPropertyType*, etc. The data about the analyzed network and the analysis results are stored in these tables.

Figure 6. Relational part of Attack Modeling and Security Evaluation Component (AMSEC) repository data scheme.



The table *Network* is for storage of the list of analyzed networks. It contains name of the network, network description, and the references to objects from the table *System*, which is intended for common description of analyzed system. The table *System* can contain more complicated specification, which includes the description of the business process (this table enables integration with other SIEM components).

The table *host* includes the list of host descriptions.

The table *objectProperty* allows storing the properties of hosts and networks.

The table *objectPropertyType* is a directory of the properties' names and types for table *objectProperty*.

The field *refTableName* can store the name of additional directory table, and the field *value* of the table *objectProperty* contains the identifier of the record from external directory. It is used, for instance, to store the vulnerability list. The additional data scheme for vulnerabilities was created to store the NVD database. The table “CVE_List” is the root table which contains all vulnerability identifiers. To make a reference between hosts and vulnerabilities, the new property (id = 19, name = “Vulnerability”, refTableName = “CVE_List”) was created in *objectPropertyType*. Thus, it is possible to add new records to the table *objectProperty* with the type identifier equal to 19, and with the vulnerability record id from the table “CVE_List” in the field *value*.

The data schema contains also the tables *metric*, *metricType*, *event*, *fieldType*, *dataType*, *eventType*, *sourceType*. The table *metric* includes various security metrics that are calculated by the AMSEC during attack graph analysis. The table *MetricType* is a directory table for the metric types. The tables *event* and *eventType* store the alerts generated by the AMSEC.

Thus, the presented data schema allows storing all information for attack graph building and analysis and to make interaction between the AMSEC and other components of the SIEM system.

6. Repository Design

The common *data repository*, provides a cross-layer integration of different components of the SIEM system [22].

According to the research and development in data repositories area [23–25], we formulate the following basic functional requirements for the SIEM data repository:

- Data storage;
- Metadata storage;
- Possibility to correct the metadata;
- Data management at different levels of detail;
- Concurrent access to data, based on privileges;
- Support of data integrity, confidentiality and consistency;
- Support of multi-version management.

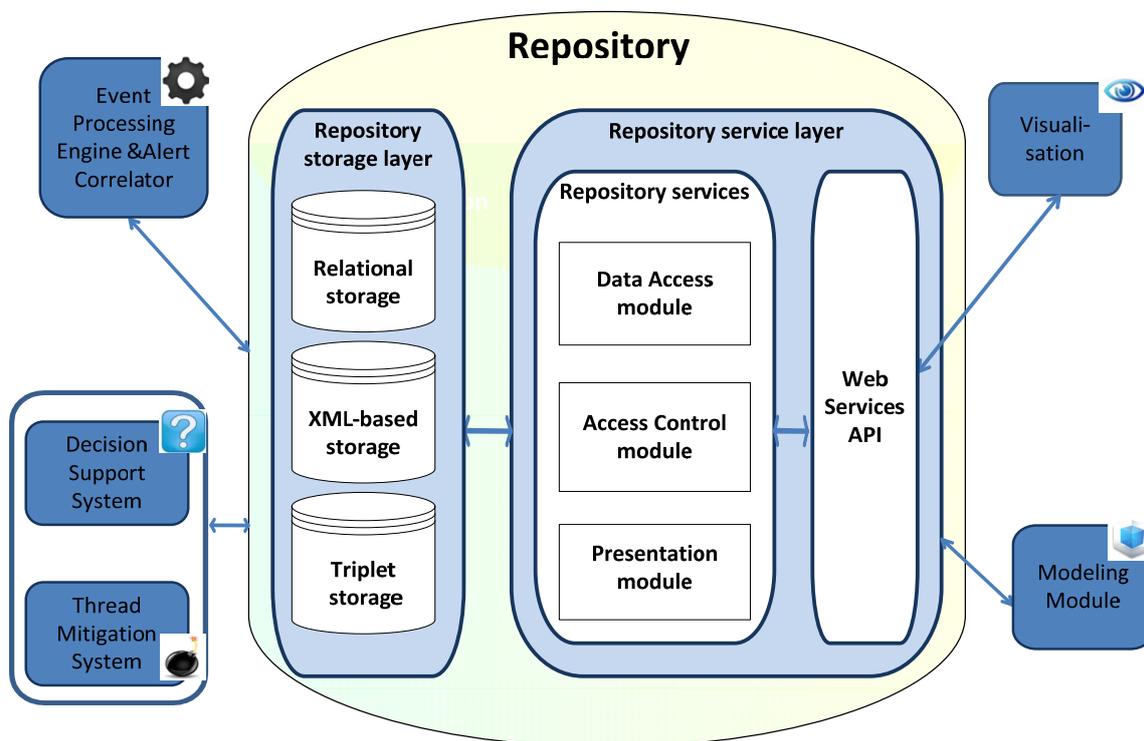
To implement the repository that satisfies the above requirements, we choose Service-Oriented Architecture (SOA), which is realized as a set of web services for data access in the repository. The advantages of this architecture are the flexibility and loose coupling of components, which provide high scalability and extensibility of the system.

SOA is a concept of the distributed information environment that joins together various software modules and applications based on well-defined interfaces and contracts between them.

The main principle of SOA is that the elements of business processes and elements of the information infrastructure, underlying them, are considered as components that are combined and repeatedly used as building blocks for the implementation of corporate processes.

Figure 7 shows the general architecture of the repository, based on SOA, and its interaction with the other components of the SIEM system (*CRUD* designates basic operations *Create*, *Read*, *Update*, and *Delete*).

Figure 7. Repository architecture.



The main components of the SIEM system are as follows:

- Event Processing Engine & Alert Correlator;
- Decision Support System;
- Threat Mitigation System;
- Visualization;
- Modeling Module, including the AMSEC.

In accordance with the main principles of SOA, the repository architecture can be divided into two basic layers:

- Repository storage layer and
- Repository service layer.

The *Repository storage layer* includes main data of the SIEM system. To support different information models developed in the MASSIF, we suggest using a hybrid approach in the repository storage layer [26].

This approach combines the capabilities of relational DBMSs and triplet stores. The triplet store provides an ontological representation of the data model, and uses logical reasoning to select data. A hybrid approach is discussed in more detail in the next section.

The *Repository service layer* allows abstracting the interaction between one or more components of the SIEM system through an intermediate interface API. It consists of three modules:

- data access module;
- presentation module and
- access control module.

The *data access module* interprets the queries for data retrieval, received from the SIEM components in the language notation used by the DBMS.

The *access control module* checks access rights of generated queries to the repository according to the permissions defined for the tables and fields in the repository.

The *presentation module* covers everything that is related to the user interaction with the SIEM system. It can be implemented as a command line, text menu, or graphical interface designed as a rich client (Windows, Swing API, *etc.*), or based on HTML. The main features of the presentation module are mapping of information and interpretation of user commands with their conversion into the corresponding operations in the context of the domain (business logic) and data source.

7. Repository Implementation

Our proposals for implementation of the hybrid SIEM data repository are, firstly, the recommendations on the choice of DBMS.

In order to choose DBMS for the repository, we investigated a set of relational databases, XML-based databases, and triplet stores. A *triple store* is a purpose-built database for the storage and retrieval of RDF metadata [27]. A *triple* is a short formal statement in the form of “subject-predicate-object”. Of course, traditional and popular relational DBMS (such as MySQL and PostgreSQL) together with XML-based DBMS can be used, but for the realization of an ontology-based SIEM, which includes advanced possibilities for logical reasoning, the triplet stores are preferable.

The storage of triplets can be divided into two basic groups [28]:

- implemented as standalone solutions (AllegroGraph, BigOWLIM and PelletDb) and
- parts of complex enterprise semantic system stores (Virtuoso, OpenAnzo and Semantics.Server).

Our analysis shows that the most preferable solution is the Virtuoso by the OpenLink Software Company [29]. It is a very powerful Enterprise-product, which has a free version and combines the support of all three types of storages that implements all necessary languages and protocols for data access and also supports a variety of necessary drivers. Also it supports such frameworks as Jena, Sesame, *etc.* Jena is a java framework for building semantic web applications. Sesame is an open source framework for processing RDF data [30].

For these reasons as the best practical solution for the ontological data storage we proposed to combine the storage of triplets and the relational databases. This provides a balance in the flexibility of data manipulation, the effective use of metadata and the acceptable processing speed.

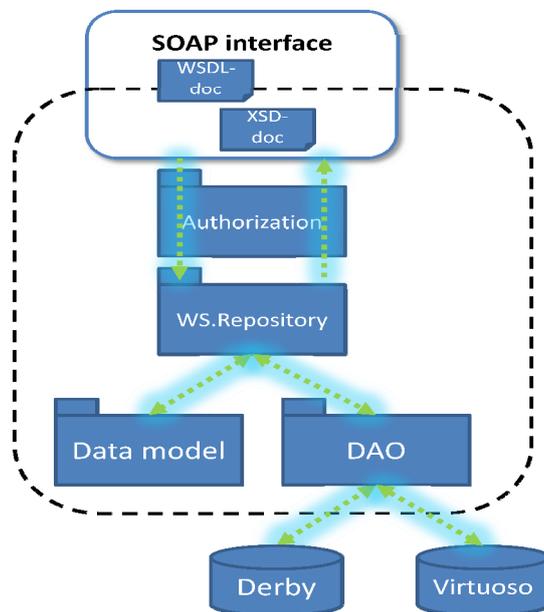
For relational databases we choose Derby and Virtuoso. Derby is used for storing events and alarms, and Virtuoso—for other security information.

The implementation of the Web services was made in Java. All Web services are implemented as stateless, *i.e.*, services do not share among themselves any variables and objects. This allows running the request from the client in a single thread on the application server. Thus, a single service can handle multiple threads of the same instances of classes.

As we decided to use a hybrid repository, a part of web services was developed for relational data representation, and the other part—for ontological one. The primary protocol we used for interaction is

SOAP. The implemented structure of the data repository is presented in Figure 8. It consists of several program modules.

Figure 8. Structure of main modules of the repository.



The *Authorization* module is responsible for authorization of the client queries that manipulate data.

The *WS.Repository* component implements the application business logic and provides interaction between other modules.

The *Data Model* module is generated on the basis of the data types defined in the corresponding *XSD-document*. It presents data base entities as Java classes.

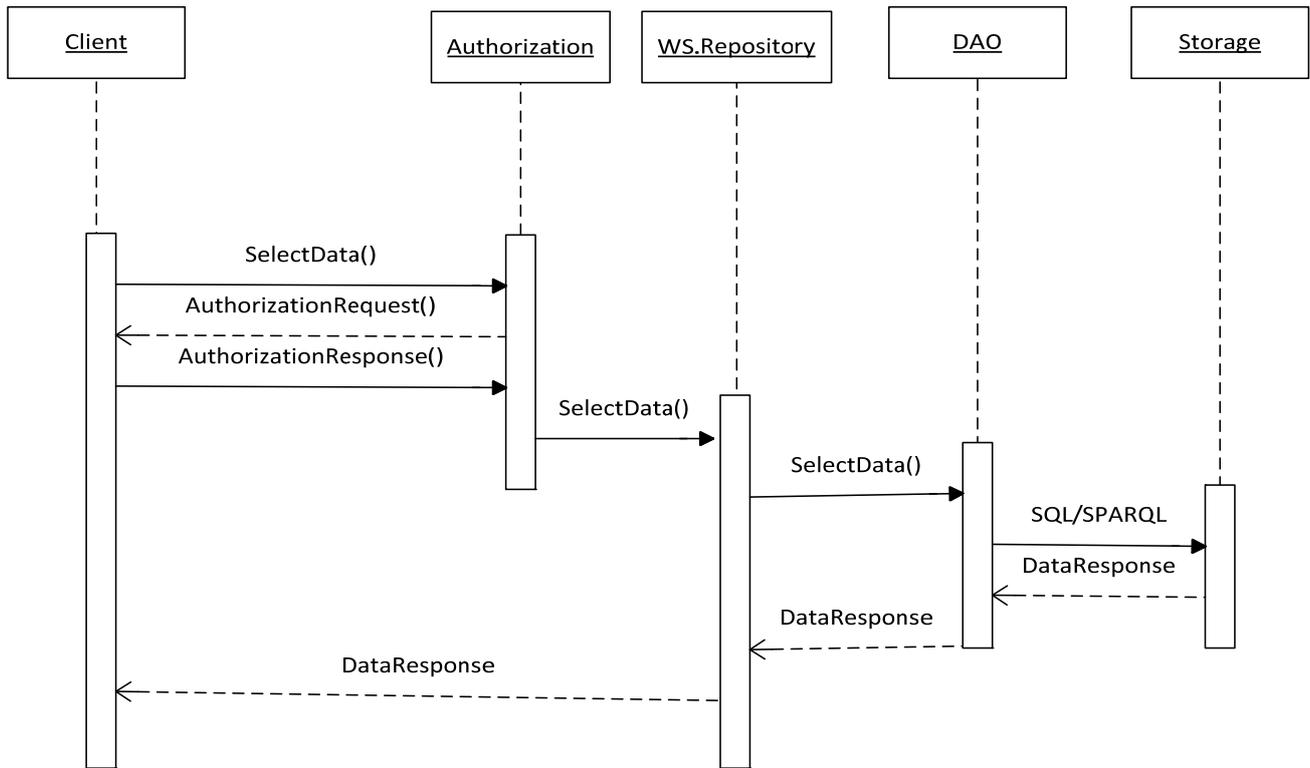
The *Data Access Object (DAO)* module translates client queries into queries in database specific language (SQL or SPARQL).

The *Web Services Description Language (WSDL) document* contains the description of the web-services implemented in the data repository.

Figure 9 illustrates the UML sequence diagram of the repository.

It shows interactions occurring when the user requests data stored in the repository. The user initiates interaction by sending request *SelectData()*. This request is processed by the *Authorization* module, which sends the *AuthorizationRequest()* backward to the user. If the client has permissions to get access to the requested data, his/her query *SelectData()* is transferred to the *WS.Repository* module. The *WS.Repository* processes the query according to the predefined logic and gives it further to the *DAO* module. The *DAO* module translates the query into *SQL* or *SPARQL* queries. The choice of the language is determined by the type of the repository the query is addressed. The *DAO* module converts the received response *DataResponse* backwards into Java objects and gives it to the *WS.Repository*, which in its turn sends *DataResponse* to the client.

Figure 9. Sequence diagram of the repository and client interaction.



The web-services are described in the WSDL document, which consists of the predefined logical parts: *definitions, types, messages, portType, binding, etc.* [31]. On the basis of the WSDL document, the client and server parts of the repository are generated.

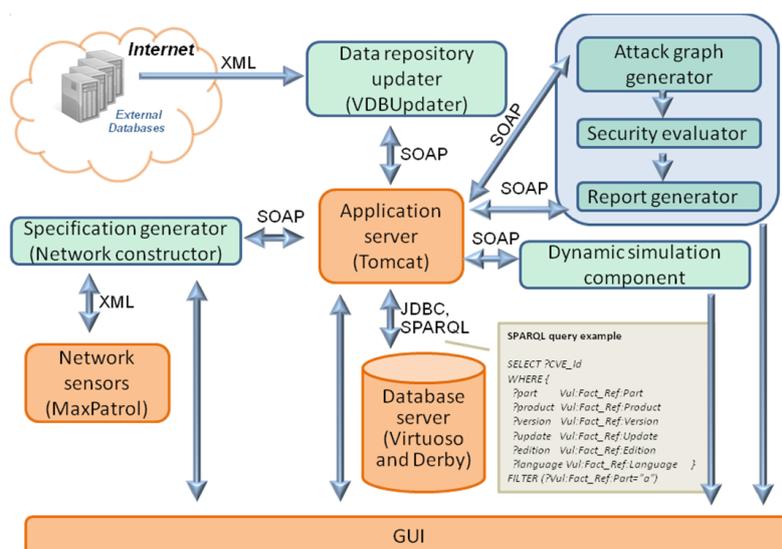
The interaction between components in the web service technology is implemented using XML-messages [32]. The SOAP specifies the XML-based format for exchanging information in the Web service implementation. As the XML-messages are the text files, different transport protocols (such as HTTP, SMTP, FTP) could be used in order to organize interaction process, though the most frequently used transport protocol in the real-world applications is HTTP. The Web Services Description Language is an XML-based language used for description of the web service functionality. It operates with definitions of ports and messages, which are separated from their implementation, allowing the reuse these definitions. The Universal Description, Discovery and Integration (UDDI) protocol provides a mechanism for publishing and locating web service applications in the Internet.

8. Using the Repository for Attack Modeling and Security Evaluation

The hybrid repository was tested for fulfilling the task of attack modeling and security evaluation by using the AMSEC.

Figure 10 depicts the interaction of the repository with other components of the AMSEC.

Figure 10. Interaction of the repository with the AMSEC.



The *data repository updater* downloads the open databases of vulnerabilities, attacks, configuration, weaknesses, platforms, and countermeasures from the external environment.

The *specification generator* converts the information about network events, configuration, and security policy, from other SIEM components or from users, into an internal representation.

The *attack graph generator* builds attack graphs (or trees) by modeling sequences of malefactor’s attack actions in the analyzed computer network using information about the available attack actions of different types, the services dependencies, the network configuration, and the used security policy.

The *security evaluator* generates combined objects of the attack graphs (routes, threats) and service dependencies, calculates the metrics of combined objects on basis of the security metrics of elementary objects, evaluates the common security level, compares obtained results with requirements, finds “weak” places, and generates recommendations on strengthening the security level.

The *reports generator* shows vulnerabilities, detected by the AMSEC, represents “weak” places, generates recommendations on strengthening the security level, and depicts other relevant security information.

The attack graph generator, the security evaluator and the reports generator are included in the *Security Evaluator* component.

The repository components, as shown in Figure 10, are the *Application server* (service implementation layer), the *Database server* (storage layer) and the *GUI* (presentation layer).

For interaction with the AMSEC we have tested two versions of the repository: (1) relational and (2) hybrid.

During functioning the AMSEC constantly requests information from the repository in order to get a list of vulnerabilities for a specific set of software and hardware installed on the simulated hosts. Data about vulnerabilities are originally loaded from the NVD database.

We performed a set of experiments aimed to estimate the time required for each step of AMSEC functioning. For experiments we used the client and server computers with the following characteristics: processor—Intel(R) Core(TM) i5, 1.80GHz, RAM—4 Gb, OS—Windows 7.

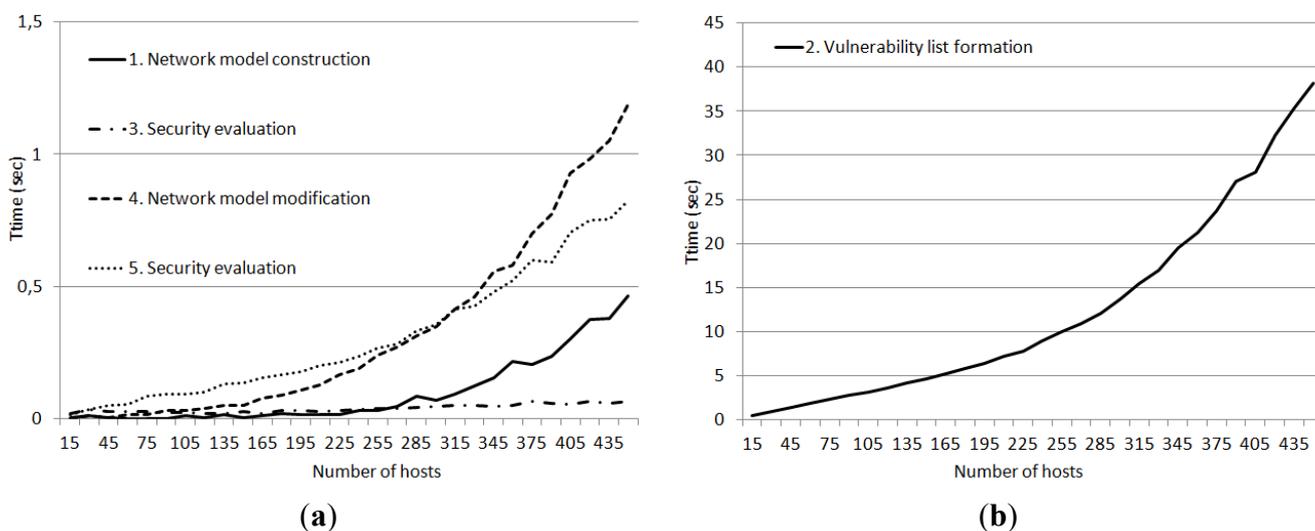
The list of steps of attack modeling and security evaluation contains the following five items:

1. Network model construction (on the basis of source data collected from the network scanner);
2. Vulnerability list formation (on the basis of software and hardware installed on each host);
3. Security evaluation (for each host and for the whole network);
4. Network model modification (according to real network changes, about 10% of all hosts were replaced);
5. Security evaluation (for changed network).

When we use the relational repository, the vulnerabilities are stored as strings with the inclusion of logical operands. Analysis of these formulas takes place on the side of the AMSEC components. This separation of functions has led to the increase in the volume of data transmitted over the network and the growth of computational costs. It has been due to the fact that all vulnerabilities have been passed on to the AMSEC, and all formulas have been tested on the side of the AMSEC components. In all, with this implementation of the repository, more than 55 thousand vulnerabilities have been transferred over the network and checked on the side of the AMSEC.

The Figure 11 shows the dependency between the time required for different steps of attack modeling and security evaluation process and the amount of hosts in the network. The source network was generated randomly with condition that each host should contain at least one vulnerable software installed.

Figure 11. Dependency between the time (sec) required for different steps of AMSEC functioning and the amount of hosts in the network (when the relational repository is used).

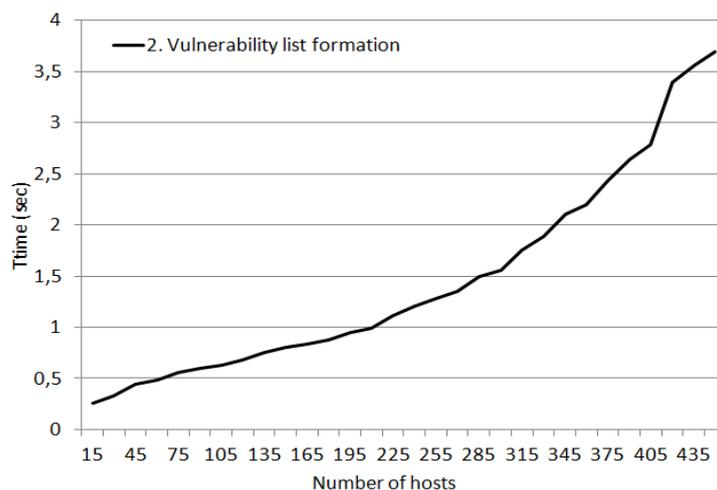


The experiments showed that the most time consumption step was the second step—the vulnerability list formation on the basis of software and hardware installed on each host. Other AMSEC steps required only few seconds (even when the network consists of 450 hosts).

When using the ontological approach, the vulnerabilities have been stored in a much more convenient form for inquiries. It has allowed analyzing the formulas on the side of the repository. Thus, in the average the sample of records under an inquiry has been shrunk to about 100 entries. This reduction has allowed significantly improving the performance of the AMSEC and reducing the load on the network channel.

Figure 12 shows the dependency between the time required for vulnerability list formation and the amount of hosts in the network. There is substantial gain in fulfilling this step in comparison with using the relational repository (see Figure 11b).

Figure 12. Dependency between the time (sec) required for vulnerability list formation and the amount of hosts in the network (when the ontological repository is used).



In general, the results of the experiments have showed that the ontological representation and the use of the hybrid repository designed in accordance with the proposed solution can reduce the necessary computing and communications resources and thereby improve the performance of SIEM systems.

9. Conclusions

In the paper we considered the task of applying the ontological approach as an addition to the relational approach for data representation and building a hybrid repository in new generation SIEM systems. We developed a hybrid (ontological and relational) repository and integrated it with the Attack Modeling and Security Evaluation Component of the SIEM system.

For these purposes, we proposed the following innovations, which are the main contributions of the paper.

First, for data representation and modeling we have proposed and applied the ontological approach that provides the necessary flexibility to the internal data representation in the repository and the possibility of using logical inference for more accurate and high-quality queering.

Secondly, we have proposed a hybrid approach to the repository implementation, which integrates the use of the relational databases and the stores of triplets.

Finally, the repository architecture has been suggested and implemented. It has been tested with the data used in the Attack Modeling and Security Evaluation Component of the SIEM system. The performed experiments have showed that the ontological data model of vulnerabilities for the AMSEC allows loading from the database much smaller amount of data and shifting the analysis task to the logical reasoning system in the repository.

In the further research it is planned to expand the ontology of vulnerabilities, as well as to add different services that provide security, including modeling and analysis of security, verification of security policies, *etc.* In addition, we plan to explore the issues of logical reasoning based on the ontological repository for countermeasure generation, as well as the development of mechanisms for data visualization.

Acknowledgments

This research is being supported by grant of the Russian Foundation of Basic Research (project #13-01-00843-a), Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (contract #2.2), State contract #11.519.11.4008 and partly funded by the EU as part of the SecFutur and MASSIF projects.

Conflicts of Interest

The author declares no conflict of interest.

References

1. Miller, D.; Harris, S.; Harper, A.; VanDyke, S.; Blask, C. *Security Information and Event Management (SIEM) Implementation*; McGraw-Hill Companies: Columbus, OH, USA, 2011.
2. Schütte, J.; Rieke, R.; Winkelvos, T. Model-based security event management. *Lect. Notes Comput. Sci.* **2012**, *7531*, 181–190.
3. Baader, F.; Horrocks, I.; Sattler, U. Description logics as ontology languages for the semantic web. *Mech. Math. Reason.* **2005**, *2605*, 228–248.
4. Herzog, A.; Shahmehri, N.; Duma, C. An ontology of information security. *Int. J. Inf. Secur. Privacy* **2007**, *1*, 1–23.
5. López de Vergara, J.E.; Villagrà, V.A.; Holgado, P.; de Frutos, E.; Sanz, I. A semantic Web approach to share alerts among security information management systems. *Commun. Comput. Inf. Sci.* **2010**, *72*, 27–38.
6. Cruz, I.F.; Gjomemo, R.; Lin, B.; Orsini, M. A Constraint and Attribute Based Security Framework for Dynamic Role Assignment in Collaborative Environments. In Proceedings of the 4th International Conference on Collaborative Computing, Orlando, FL, USA, 13–16 November 2008; pp. 322–339.
7. Kolovski, V.; Hendler, J.; Parsia, B. Analyzing Web Access Control Policies. In Proceedings of the 16th international Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 677–686.
8. Rochaeli, T.; Eckert, C. RBAC Policy Engineering with Patterns. In Proceedings of the Semantic Web and Policy Workshop, Galway, Ireland, 7 November 2005.
9. Fitzgerald, W.M.; Foley, S.N.; O’Foghlu, M. Confident Firewall Policy Configuration Management using Description Logic. In Proceedings of the Twelfth Nordic Workshop on Secure IT Systems, Reykjavik, Iceland, 11–12 October 2007.

10. Taylor, K.; Leidinger, L. Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. *The Semantic Web: Research and Applications*. In Proceedings of the 8th Extended Semantic Web Conference (ESWC'11), Heraklion, Greece, 29–30 May 2011; pp. 285–299.
11. Razzaq, A.; Ahmed, H.F.; Hur, A.; Haider, N. Ontology Based Application Level Intrusion Detection System by Using Bayesian Filter. In Proceedings of 2nd International Conference on Computer, Control and Communication (IC4), Karachi, Pakistan, 17–18 February 2009; pp. 1–6.
12. Rochaeli, T.; Eckert, C. Attack Goal Generation Using Description Logic-Based Knowledge Representation. In Proceedings of the 2005 International Workshop on Description Logics (DL2005), Edinburgh, Scotland, UK, 26–28 July 2005.
13. Schatz, B.; Mohay, G.; Clark, A. Generalizing Event Forensics across Multiple Domains. In Proceedings of the 2nd Australian Computer Network & Information Forensics Conference (Forensics 2004), Edith Cowan University, Perth, Australia, 25 November 2004; pp. 136–144.
14. Kenaza, T.; Yahi, S.; Benferhat, S. From representing contextual intrusion detection information in description logics to monitoring target events. *Agence Natl. Rech. Délivr.* **2006**, *10*, 1–19.
15. Nicolett, M.; Kavanagh, K.M. *Critical Capabilities for Security Information and Event Management*; Gartner RAS Core Research Note G00 212420; Gartner: Stamford, CT, USA, 2012.
16. Ogle, D.; Kreger, H.; Salahshour, A.; Cornpropst, J.; Labadie, E.; Chessell, M.; Horn, B.; Gerken, J.; Schoech, J.; Wamboldt, M. *Canonical Situation Data Format: The Common Base Event V1.0.1*; International Business Machines Corporation: Armonk, NY, USA, 2004.
17. Common Event Format. Available online: http://www.arcsight.com/solutions_cef.htm (accessed on 25 January 2013).
18. Curry, D.; Debar, H. Intrusion detection message exchange format data model and extensible markup language (XML) document type definition. Technical report, IETF Intrusion Detection Working Group, 2003. Available online: <http://www.ietf.org/proceedings/50/I-D/idwg-idmef-xml-03.txt> (accessed on 25 January 2013).
19. Common Information Model (CIM), DMTF. Available online: <http://dmtf.org/standards/cim> (accessed on 25 January 2013).
20. Security Content Automation Protocol (SCAP). Available online: <http://scap.nist.gov> (accessed on 25 January 2013).
21. Kotenko, I.; Chechulin, A.; Novikova, E. Attack Modelling and Security Evaluation for Security Information and Event Management. In Proceedings of the International Conference on Security and Cryptography (SECRYPT 2012), Rome, Italy, 24–27 July 2012; pp. 391–394.
22. Kotenko, I.; Polubelova, O.; Saenko, I. Data Repository for Security Information and Event Management in Service Infrastructures. In Proceedings of 9th International Joint Conference on e-Business and Telecommunications (ICETE 2012). International Conference on Security and Cryptography (SECRYPT 2012), Rome, Italy, 24–27 July 2012; pp. 308–313.
23. Garcia-Molina, H.; Ullman, J.D.; Widom, J.D. *Database Systems. The Complete Book*, 2nd ed.; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2009.
24. Marco, D. *Building and Managing the Meta Data Repository: A Full Lifecycle Guide*; Wiley: Hoboken, NJ, USA, 2000.
25. *Triple Store Evaluation Analysis Report*; Revelytix Inc.: Sparks, MD, USA, 2010.

26. Kotenko, I.; Polubelova, O.; Saenko, I. The Ontological Approach for SIEM Data Repository Implementation. In *Proceeding of the 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing*, Besançon, France, 20–23 November 2012; pp. 761–766.
27. Barret, R. XML Database Products: Native XML Databases, 2010. Available online: <http://www.rpbouret.com/xml/ProdsNative.htm> (accessed on 25 January 2013).
28. Storage and Inference Layer Solutions. Available online: <http://alexidsa.blogspot.com/2009/12/sail.html> (accessed on 25 January 2013).
29. Virtuoso. Available online: <http://virtuoso.openlinksw.com> (accessed on 25 January 2013).
30. Comparison of Triple Stores. Available online: http://www.bioontology.org/wiki/images/6/6a/Triple_Stores.pdf (accessed on 25 January 2013).
31. Web Services Description Language (WSDL) 1.1. Available online: <http://www.w3.org/TR/wsdl> (accessed on 25 January 2013).
32. Web Services. Available online: <http://www.w3.org/2002/ws/> (accessed on 25 January 2013).

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).