ITC 4/47
Journal of Information Technology and Control
Vol. 47 / No. 4 / 2018
pp. 655-667
DOI 10.5755/j01.itc.47.4.20007

An Ontology-Based Storage of Security Information

Received 2018/01/22 | Accepted after revision 2018/10/15

http://dx.doi.org/10.5755/j01.itc.47.4.20007

An Ontology-Based Storage of Security Information

Igor Kotenko, Andrey Fedorchenko, Elena Doynikova, Andrey Chechulin

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS);
14-th Liniya, 39, St. Petersburg, Russia; e-mail: { ivkote, fedorchenko, doynikova, chechulin}@comsec.spb.ru
St. Petersburg National Research University of Information Technologies, Mechanics and Optics
(ITMO University); 49, Kronverkskiy prospekt, Saint-Petersburg, Russia

Corresponding author: ivkote@comsec.spb.ru

The paper suggests an ontology-based approach for design of security data storage. It analyzes heterogeneous security information for construction of the security storage and the statistics of links between various security data sources. The suggested ontological model of the data storage allows connecting both heterogeneous security data and security data of the same type from various sources. The main features the ontological model, its key elements and links between them are described in details. In addition, the authors introduce the developed prototype of the ontological data storage and provide examples of its usage for security evaluation.

KEYWORDS: ontology, security data, data sources, data analysis, security information storage.

1. Introduction

One of the most important research directions in information security is the development of security information and event management (SIEM) systems. The significant element of such systems is a security repository which stores all data required for security information analysis and decision making. The sources for this data can be divided into two groups – external and internal (for the system under protection). The internal sources include the data from intrusion detection and prevention systems, antiviruses, network scanners, software logs and others. These sources provide the main part of data for security information analysis. In addition, there are also external sources that contain data that helps to perform the analysis of internal security information. These sources can include the common databases of vulnerabilities, attack patterns, hardware and software as well as many others.

The global challenge of this information usage for various security areas occurs in a big number of het-

erogeneous data sources that should be collected and processed. This circumstance essentially impedes the efficient use of the security information. In turn, it affects the general security level of computer infrastructures. One way to address this challenge is to integrate these data sources taking into the account the diversity and absence of links between them. Research and development in this area has been under way for many years, but at the moment there are no comprehensive solutions in the open access. The current status is caused by a multitude of factors, including the usage of proprietary formats to describe the security data by commercial companies and research organizations usually, the lack of techniques for linking of disparate security data, inconsistent filing of security data storages and others.

In the paper, we analyze several kinds of security information sources and propose the approach for their integration in a unified storage by applying the suggested ontological approach. We suppose that the resulting storage will allow not only to find information (vulnerabilities, attacks, exploits etc.) based on the security scanners reports, but also to generate new knowledge about the system using analysis of the current security situation and interrelationships between information objects in the ontological storage.

The novelty of the paper lies in the proposed technique of the ontological inference which is based on the proposed ontological data model and the security data storage implementation. In the future, this technique will allow us to develop a new generation of intelligent systems for security information and event management. Currently, this research is especially relevant due to continuous growth in the amount of various security data sources, expansion of the areas of security related applications (for instance, Internet of Things, social and cyber-physical systems), relations between security related objects, etc. However, it is supposed that the ontological approach allows us to use more accurate queries and thus reduce the required time for query processing. This advantage is particularly important in the field of distributed network security management, because there is a need to perform the in-depth analysis of heterogeneous information including historical records.

Therefore, the general goals of the suggested security information storage based on the ontological data model are as follows: (1) integration of the heterogeneous data from various sources; (2) search of implicit relationships between security information of the same and various types; (3) extraction of the new knowledge based on the logical inference; and (4) automated usage of gathered knowledge for information security management.

This paper is an extended version of the paper presented at the 11th International Symposium on Intelligent Distributed Computing (IDC 2017) [20]. In the present paper, we extended related works section, improved the ontological model and added its detailed description, we extended the description of the prototype, and finally we described the experiments with the enhanced ontological data storage while in the previous version we just approved that this storage can be constructed.

The contribution of the paper can be summarized as follows: (1) the extended ontological model that allows connecting both heterogeneous security data and security data of the same type from various sources is constructed and described in details; (2) the developed prototype of the ontological data storage is introduced; and (3) the experiments that show the prototype operation are provided (namely, examples of the prototype usage for the logical inference).

The paper is organized as follows. Section 2 overviews the sources of security information and the existing approaches aimed at integrating this information. Section 3 contains the description of the ontological security data storage and inference techniques that are able to extend the available knowledge. Section 4 outlines the implementation of the developed ontology. Section 5 shows the interrelationships of the available databases and results of the experiments with the ontological security storage. Finally, conclusion describes the obtained results and the future research plans.

2. Security Sources and Related Work

Over the past decades, the growth of the interest to the complex security monitoring has resulted in generation of a multitude of different security databases, including the databases of vulnerabilities, weak places, (vulnerable) configurations, exploits, platforms, attack patterns, remediation, malware, software and hardware updates, network traffic, security events and many others.

In the paper, we analyze the first seven aforementioned data types used in the most extensive and detailed databases.

Vulnerability databases are one of the oldest sources of security information. To date, more than 100 000 vulnerabilities for more than 60 000 hardware and software products have been found. Vulnerability databases, as databases of a particular type of security data, are used in vulnerability scanners [29], Web application firewalls [32], and also applied in conjunction with other security information to evaluate network infrastructure security by attack modeling [1, 18, 19] and risk assessment [9, 21].

The best known vulnerability sources are as follows: "Common Vulnerabilities and Exposures" (CVE) [6], "National Vulnerabilities Database" (NVD) [26], "Open Source Vulnerabilities Data Base" (OSVDB) [28], "Vulnerability Notes Database" (VND) [38], SecurityFocus project with BugTraq [33], and IBM X-Force [16].

Dictionaries of software and hardware (platforms) products are important security data, as they enable to identify potentially vulnerable objects. At the moment, there is only one open, standardized and accessible dictionary a Common Platform enumeration (CPE) developed by MITRE. The format of the product dictionary of the Common Vulnerability Reporting Framework (CVRF) is a successor of the CPE format. The main feature of this format is a hierarchical structure of product names which provides a more understandable representation and the ability to uniquely identify records. The product records are stored in the dictionaries "Common Platform Enumeration" (CPE) [4] and "Common Vulnerability Reporting Framework" (CVRF) [17].

Exploit databases contain specifications of software, files, requests, or a sequence of commands that use vulnerabilities in order to cause unintended or unanticipated behavior. Exploits are often used during the penetration testing, malicious attack performing and malware activity. However, they can be also used in attack recognition process if a security system detects the presence of an exploit or its traces, it usually generates alarms and starts the actions aimed on attack prevention. One can say that these databases are the most practical among the all types of security information sources. The main source with information about exploits is "Exploit DataBase" (EDB) [27]. These databases should be integrated to maximize the efficiency of their application. The most popular and comprehensive project in this area is Metasploit [25].

Attacks patterns are also very important security data aimed at monitoring and protection of distributed networks. The basis of the attack pattern specification consists in description of attack implementation methods, attack steps and attack techniques, as well as fields that refer on the exploited vulnerabilities and weaknesses. This information can also be obtained in other formats from various sources. For example, it can be attack patterns from specific intrusion detection systems. This kind of security information can be found in the "Common Attack Pattern Enumeration and Classification" (CAPEC) database [2].

Weaknesses of software and hardware are represented by using a classification of vulnerabilities. Thus, the presence of a weakness indicates a potential vulnerability, and the presence of a vulnerability is the direct evidence of a weakness. Information about weaknesses available in the "Common Weakness Enumeration" (CWE) database [8].

Remediation databases are valuable sources of security data; they are valuable for countermeasure generation. One of the formats to represent countermeasures is "Common Remediation Enumeration" (CRE) [5].

Configuration databases contain descriptions of recommended secure settings for specific software platforms. Usually, these settings are defined by software developers on the basis of their experience and best practices. However, currently, the application of this opportunity is limited due the absence of the CRE database. The main source of information about configurations is the "Common Configuration Enumeration" (CCE) database [3].

One of the main challenges while drawing up a common picture of a security situation is both the integration of one data type from different sources and the integration of different data types among themselves. This challenge has been investigated for more than 10 years. For example, the process of vulnerability databases integration is outlined in [11, 34, 36].

In case of integration of security data of different types the most valuable for the community from our

point of view are those security databases that have links with other databases or information objects. It helps security specialists to create an interconnected understanding of the security information. In this paper, this aspect is reviewed in details in the next sections. In addition, when security data of different types are combined, the challenge of integration of security data of the same types is still essential, excluding the case when the data of specific type are gathered from the same source. Consequently, when a few sources of data of the same type (for example, different vulnerability databases) are used, there is a data inconsistency challenge. In this case, the data should be preprocessed to transform the data on the same objects from different sources to one format.

An integration of the security data of different types was considered by the authors earlier in [22]. In [22], the drawbacks for application of relational databases compared to ontological databases for security monitoring were also outlined. This is because in the case of security monitoring, the logical inference based on available data as well as not labor-consuming modification of the data model are essential functionalities. Thus, one of the well-proven methods for the security data integration from different sources is an ontological approach. The ontological approach as a whole and the tools for construction of ontologies are well described in [15, 31, 39].

In [10, 14, 41], the vulnerability-centric ontologies for security analysis are presented. In [23], the common approach to the ontological storage generation is introduced that we evolve in this paper. In [24], a security metrics ontology for security assessment is suggested. In [13], an ontological data model is used to represent the information used by SIEM systems and the operations with this information. The developed model is used to implement the logical inference for countermeasure generation. The paper [30] deals with the construction of a common ontology for the SCAP protocol [40]. The SCAP protocol includes the following types of security data: vulnerabilities [6], configurations [3], software and hardware [4], etc. The main goal of this protocol is the integration of security data.

Special attention should be given to an Unified Cybersecurity Ontology (UCO) presented in [35, 37]. UCO is an ontological model that integrates different security data for security assessment. This ontology uses such standards as CVE, CWE, CAPEC, CCE, and others, to specify different security entities. This model allows to specify different information and communication entities, namely, files, network addresses, processes, operation systems, etc. Other useful entities are entities of network state and attackers. However, UCO cannot integrate information from various sources of the same type, because it does not contain appropriate connecting properties. In addition, the used storage requires the mandatory configuration during implementation, because some concepts are specific for the security monitoring of particular infrastructures. Furthermore, the entities that represent the current system state in real time are hard to support because it requires an operative modification of an ontological database.

It should be noted that in spite of the large amount of the security databases, they do not allow to form a concerted common security picture due to the disunity of these databases. Their joint application is a rather complex task and requires high costs on pre-processing. From our point of view, there is a need in construction of the common security data storage. This storage should allow operative processing and modifying of the security information. An ontological approach is well suitable for the integration of databases. Opposite to the relational database, it allows to implement the flexible logical inference on the basis of available data and the simple data model modification.

3. An Ontological Model of the Data Storage and Inference Technique

As it was noted, the main disadvantage of the existing databases lies in impossibility to form the common security picture, because of disunity of these databases. An ontological approach is one of the solutions to represent the interconnected data to process the data of complex structures. It allows to express complex relations between entities using description logics. The approach consists in definition of the set of concepts in the selected subject area. Then, the connections between the concepts are generated considering their relations and interactions. As it was shown in the previous section, the papers suggest various ontologies for the security related data. However, to the best of our knowledge, there is no ontological storage that incorporates data from different databases considering the nature of the relationships for these data.

This paper presents the developed ontological model. The concepts are defined considering the next information objects: vulnerabilities, software, software weaknesses, exploits, attack patterns, software and hardware configurations as well as remediations. To identify the structure of interconnections between these objects, we reviewed the main open databases and outlined the relations between them.

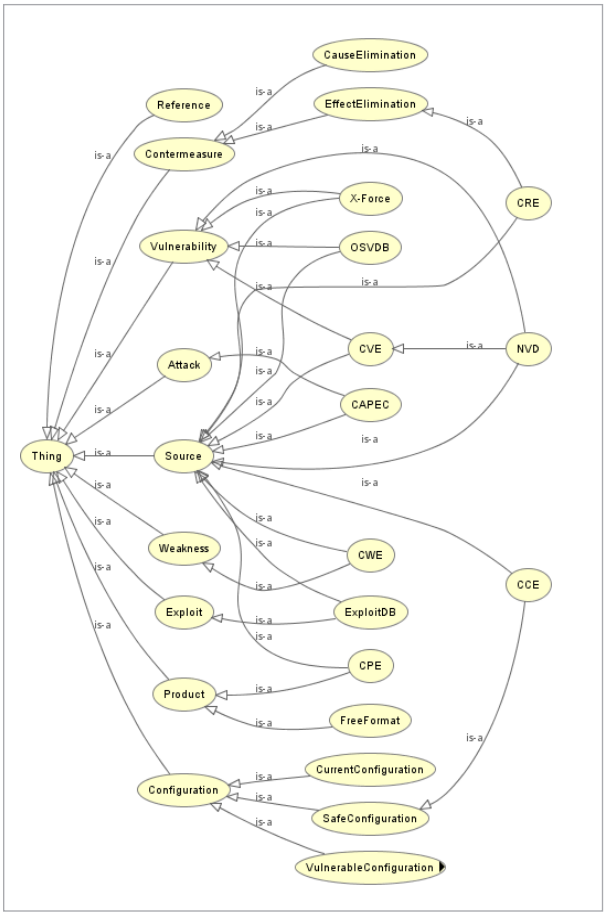
Figure 1 depicts the common class inheritance hierarchy of the ontological model for the security data storage. In the figure, nodes denote the classes of security information ("*Vulnerability*", "*Attack*", "*Exploits*", etc), their sources ("*CVE*", "*CAPEC*", "*ExploitDB*", respectively) and the individual security information elements (links to sources, current infrastructure configurations, countermeasures to eliminate the causes of threats) specially allocated to the relevant classes ("*Reference*", "*CurrentConfiguration*", "*CauseElemination*"). The arcs (denoted by "is-a" relations) are relations between classes. The direction of the relationship determines the relation "ancestor-successor" and can be interpreted as "the successor class is a subclass of the ancestor class". For example, the class "*CAPEC*" is a subclass of the classes "*Attack*" (attack pattern) and "*Source*" (source of security information); and the class "*NVD*" is a subclass of the classes "*CVE*", "*Vulnerability*" and "*Source*".

The root of the hierarchy ("*Thing*" entity according to the Web Ontology Language OWL2) is the security information. All classes are conditionally divided on: (1) security information types (excluding the "*Source*" and "*Reference*" classes and their inheritors); (2) security information sources ("*Source*" and child classes); (3) references to the third-party sources of specific security information ("*Reference*"). The provided model describes both the parental relationships between the concepts of the "*Security information*" knowledge field and the membership relationships between the concepts (types) and the specific data sources. However, the class inheritance hierarchy does not represent relationships between entities (properties-objects) and possible variants of description of their individuals (properties-values).

In the previous work [12, 20], it was fixed that the class of vulnerabilities has the largest number of links with other classes of security information within the ontological model. Thus, we start the specification of object properties from this class.

Figure 1

The common class inheritance hierarchy of the ontological model for the security data storage



We outline four so-called *irrational properties* (according to [31, 39]):

1. "*implementedBy*",
2. "*implementedIn*",
3. "*implementedIf*", and
4. "*implements*".

The irrationality of these properties-objects consists in the limitation of the instances: these instances cannot use these relationships to themselves. In other words, only two different individuals can be connected by the above properties.

The following axioms that are based on the listed above object properties are valid for the "*Vulnerability*" instance:

1. "*implementedBy Exploit*";

2 "*implementedIn Platform*";

3 "*implementedIf VulnerableConfiguration*";

4 "*implements Weakness*";

5 "*implements Attack*".

Thus, the specified properties describe the relations between the vulnerabilities and five other information types.

In addition, there are relationships between the instances of security information classes of the same type and the ones of different types that do not connect concepts directly (using direct relationship). This object property describes the relationship of the object with some external source (for example, via URL or any other conditionally unique identifier), and it is represented with the "*Reference*" instance.

Thus, the individuals in the security storage have object property "*connectedWith*". The "*Reference*" class instances are objects of this statement and they define the range of allowed values. The specified property is *symmetric* as soon as the relationship between an individual and an external source is equivalent to the relationship between the external source and individual. The last statement allows connecting the individuals of the security storage of the same type and of the different types using the "*Reference*" entity. The common enumeration of the properties-objects of the developed ontological model and their specific characteristics are provided in table 1.

In Table 1, the object properties of the suggested ontological model are depicted, where *T*, *S* and *I* denote transitivity, symmetry and irrationality, accordingly [31, 39].

When an object property is *transitive,* it means that if it connects individuals a and b, and it also connects individuals b and c, we can conclude that it connects individuals a and c. For example, for the axioms "*Vulnerability implementedBy Exploit*" and "*Weakness implementedBy Vulnerability*" the following statement is valid: "*Weakness implementedBy Exploit*".

The description of the irrationality and symmetry was provided above in the text.

Some entities that describe individuals within the ontological model are implemented using the specification of data properties. These properties are the fields that specify security information.

For example, Figure 2 (left) represents the hierarchy of security metrics for the Common Vulnerability Scoring System (CVSS) of version 2.0 [7] that is used to describe vulnerabilities. In this case, an opportunity to specify the range of values for the data property is an essential feature of the OWL2 language. For example, the range of possible values for the metric "*AccessComplexity*" is specified using the following expression (in Protege 5.0): *{"High", "Low", "Medium"}*.

Figure 2 (right) shows the example of the description hierarchy for the "*Platform*" concept considering CPE of version 2 [4].

OWL2 language also allows one to specify the range of values both for the object properties and data properties in the form of one or several classes (domains).

For the examples in Figure 2 (specification of the CVSS metric and CPE entry) the "*Vulnerability*" and "*CPE*" classes are set as domains, accordingly.

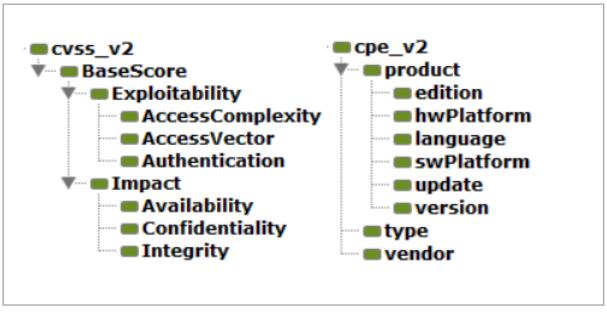
Table 1

Object properties of the ontological model for the security information storage

Object property	InverseOf	T	S	I
implementedBy	implements	+	-	+
implements	implementedBy	+	-	+
implementedIn	containsImplementationOf	+	-	+
containsImplementationOf	implementedIn	+	-	+
implementedIf	leadsToImplementation	-	-	+
leadsToImplementation	implementedIf	-	-	+
connectedWith	-	+	+	+

Figure 2

The specification of the data properties within the ontological model of the security information storage



Another powerful tool of the ontological approach is an opportunity to specify an equivalency between entities. In other words, the equivalence characteristic can be used to specify relationships between classes, object properties and data properties. For example, the text description of the vulnerability in the NVD database is provided in the "*Description*" field. In the X-Force database, the "*Overview*" field contains information of the same type (equivalent information). In some conditionally unknown database, this information is contained in the "*Annotation*" field. Thus, any aforementioned notion can be used when specifying the equivalent relationship between the provided data properties and analyzing the individuals from different databases within the ontological model.

Another possibility that requires attention is specifying the quantitative limitations (cardinality) of sets of used objects for the object properties. This mechanism involves assignment of a value for the specific property usage by the class individual and (or) assignment of the number of the individuals connected via this property.

For example, statements "*Every Exploit implements minimum 1 Vulnerability*" and "*Every Vulnerability implementedIn minimum 1 Platform*" consider each subject ("*Exploit*", "*Vulnerability*") connected with minimum one object ("*Vulnerability*", "*Platform*") using appropriate properties. In fact, an individual of the "*Exploit*" class cannot exist without an individual of the "*Vulnerability*" class (for the second statement the same axiom is valid). However, due to inconsistency of the security information sources the cardinality characteristic of properties was used carefully.

For example, an assumption that any vulnerability uses some weakness looks logical. However, not all vulnerabilities can be connected with weaknesses database because there are no appropriate direct and cross references (identifiers) in the analyzed sources. In this case, we specified cardinality using the following statement: "*Some Vulnerability implements maximum 1 Weakness*".

This construction is more flexible than fixed cardinality limitation. It allows one to exclude errors of the inadequate description of individual while using ontological model of the security information storage.

We developed the proposed ontology for application in security assessment and countermeasure selection process. To get new knowledge, on the basis of the developed ontology we suggest to use the following *ontological inference technique*.

Step 1. Collection of available security data.

It can be static data (software and hardware, vulnerabilities gathered by network scanners) and dynamic data (obtained, for example, from intrusion detection and prevention systems).

Step 2. Extension of security data using relations between ontological concepts.

It is performed bypassing security data instances using links. In the previous work [20], we outlined "strong" and "weak" relations. Link is "strong" if the ancestor concept instance uniquely indicates existence of the descendant instance. Link is "weak" if the ancestor concept instance does not guarantee existence of the descendant concept instance.

On the current level of implementation we redefined the terminology of these relations using cardinality property: link is "strong" if the number of connected individuals is minimum 1, otherwise link is "weak". If link is "strong", we proceed bypassing and put the appropriate concept instance to the output dataset; if link is "weak" we stop bypassing by this link and put the appropriate concept instance to the dataset for additional analysis.

Step 3. Data analysis.

In this stage, we analyze the output dataset (obtained via "strong" links) and dataset for the additional analysis (obtained via "weak" links). Processing of the dataset for additional analysis includes analysis of particular fields of the concept instances.

For example, the link between the software in the CPE format and the vulnerabilities in the CVE format is "strong". It means that the existence of the software instance in the system uniquely indicates existence of the vulnerabilities of this software.

On the other hand, the link between a CAPEC attack pattern and vulnerabilities in the CVE format is "weak". It means that detection of the appropriate attack pattern does not determine existence of the linked vulnerabilities. In addition, we can analyze CAPEC attack pattern fields related to vulnerable software and make more accurate conclusions.

The separation of step2 and step3 is virtual: they both are implemented during the security data instances' bypassing, but the mechanism of processing of instances' properties is different.

4. Prototype of the Data Storage

The ontology model of the security information storage was implemented using OWL (Web Ontology Language) of version 2 and description logic of type DL (Descriptive Logic).

Selection of the DL type is stipulated by its expressiveness without loss of computation completeness (all conclusions will be computable guaranteed), and with computation resolvability (all computations will be completed at a certain time).

There are some limitations: a class can be a private property, and a property cannot be an individual or a class. In its turn, the OWL profile selection is not mandatory in this stage. However, the decision must take into account both the number of classes and individuals in the model and processing time for typical requests. From our point of view, the most suitable profile is Rule Languages (RL).

The developed ontological model is oriented at the practical application of the storage based on this model. Thus, the basic predicate for the construction of the axioms about security information is exploitation (implementation) of the described entities. Moreover, the flexible model structure allows adding new data sources and security information types without modification of the already existing statements.

As it was mentioned above in [20], we introduced "weak" and "strong" relations. In this paper, we redefined the terminology of these relation using cardinality property (see Section 3).

Table 2 provides the resulting picture for the connectivity of the ontological model concepts.

The relationships between the instances of classes, when implemented using the RDF (Resource Description Framework), can be expressed as triplets [29]: subject-predicate-object. In the presented table, the left column and the upper row, denoting the classes of security information, are the sets of subjects and objects, respectively. Each cell, at the intersection of the subject-object pair, is a predicate expressed by the object properties of the ontological model. For each subject-object relationship in cells, the cardinality of the set of objects is indicated. For example, the entity "*Vulnerability*" (subject) can be "*implementedIn minimum 1*" (predicate) for the "*Platform*" object. It can also "*implement some*" "*Attack*" and "*implement maximum 1*" "*Weakness*".

If there is no relation between entities, then the corresponding cell at their intersection in the table is filled with the symbol "-".

Thus, from the one hand, Table 2 considers the limitation of the objects set cardinality for specific properties.

Table 2

Types of the relations between main ontological concepts

	Platform	Vulnerability	Attack	Weakness	Exploit	Configuration
Platform	-	containsImplementationOf some	-	-	implements some	-
Vulnerability	implementedIn minimum 1	-	implements some	implements maximum 1	implementedBy some	implementedIf minimum 1
Attack	-	implementedBy some	-	implementedBy some	-	-
Weakness	-	implementedBy some	implements some	-	-	-
Configuration	implementedBy minimum 1	-	-	-	-	-
Counter-measure	implementedIn minimum 1	-	-	-	-	implementedIn some
Exploit	implementedBy minimum 1	implements minimum 1	-	-	-	-

On the other hand, this picture is generated considering that each upper level class of the security information directly (explicitly) relates to other upper level classes via only one pair of object properties (direct and inverse).

It should be noted that in the provided table, the fields are filled only if concepts are directly connected via an object property. In other words, relations that are defined using logical inference on the basis of the properties transitivity and symmetry are not considered.

Moreover, for development of the ontological model the relations between objects were specified both on the basis of theoretical understanding of the concepts' nature and the practical exploitation of security information sources.

5. Experiments and Discussion

At the moment, the unified vulnerability database is implemented using PostgreSQL and Java [11], and the prototype of the security storage that incorporate data of different types on the basis of the proposed ontology is implemented using Virtuoso Server from OpenLink.

Let us describe two groups of the experiments performed with the unified vulnerability database and the security information storage.

The goal of the first group of experiments is analysis of the existing security related databases to validate the possibility to extend security knowledge using interrelationships between the concepts.

The second group of experiments is devoted to the specific case studies for derivation of new knowledge using interrelations between information objects.

To create the data scheme of the security information storage we collected the statistical data of interrelationships between databases (Table 3).

Table 3 shows the following characteristics:

1. the name of the database (B2) which is referenced in the analyzed (targeted) database (B1);
2. the total number of references to B2 in B1;
3. the number of B1 entries that have references to B2;
4. the number of references to unique B2 elements in B1;
5. the percentage of B1 elements that have references to B2;
6. the percentage of unique references to B2 in B1 (relation of value in column (2) to value in column (4));
7. the average number of references to B2 from one B1 element;
8. the exploitability of B2 database in B1 database (i.e. relation of number of references to B2 elements in B1 to the total number of elements in B2).

Table 3
Statistical characteristics of links between data sources

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
NVD (84557)							
EXPLOITS	2752	2599	2409	3,1%	87,5%	0,03	7,21%
CPE	2048178	82740	199160	97,9%	9,7%	24,22	169,97%
CWE	51131	50519	88	59,7%	0,2%	0,60	12,45%
CAPEC (528)							
CVE	69	36	57	6,8%	82,6%	0,13	0,07%
CWE	964	234	241	44,3%	25,0%	1,83	34,09%
EXPLOIT-DB (33394)							
CVE	33993	33993	16879	100,0%	49,7%	0,99	19,96%

Note that the reference from the analyzed database to itself in the table (for NVD-CVE, CAPEC, EXPLOIT-DB) specifies the total number of entries in the appropriate database.

We used the last versions of the security databases to analyze target and relevant data sources (April 17) except exploits database (Jan. 15).

Low level of the exploitation (column 8) for the CVE database is caused by the fact that vulnerabilities in CAPEC database are provided only as examples of vulnerabilities that can be used for the attack implementation.

In its turn, extra-exploitability of references to CPE from NVD is related with: (1) a significant extension of the unique entries of the CPE dictionary through the NVD; (2) the features of the CPE specification that lead to the impossibility of the unique identification of the vulnerable products in some cases.

Furthermore, the high level of connectivity (column 5) should be noted for: (1) NVD(CVE) database with CPE dictionary and CWE database, (2) CAPEC database with CWE database, and (3) ExploitDB database with NVD(CVE) database.

The results obtained for analysis of the relations between the data sources approve possibility of extension of the security knowledge using logical inference on the basis of the developed ontology.

To demonstrate the possibility of new knowledge generation, using the developed security information storage, we prepared an example.

This example shows the analysis of heterogeneous security information based on ontological data model rules (see Figure 3).

The example demonstrates the technique of interrelationships analysis based on cross links processing in action. Here it should be noticed the "*connectedWith*" property which is high-level in a hierarchy of properties that defines relationships between the ontology classes. Thus all objects' properties of ontological model ("*isImplementedBy*", "*implements*") may be generally represented by this property.

The example shows the links between security information in the source data. By querying the ontological model, it is possible e.g., to answer the following questions:

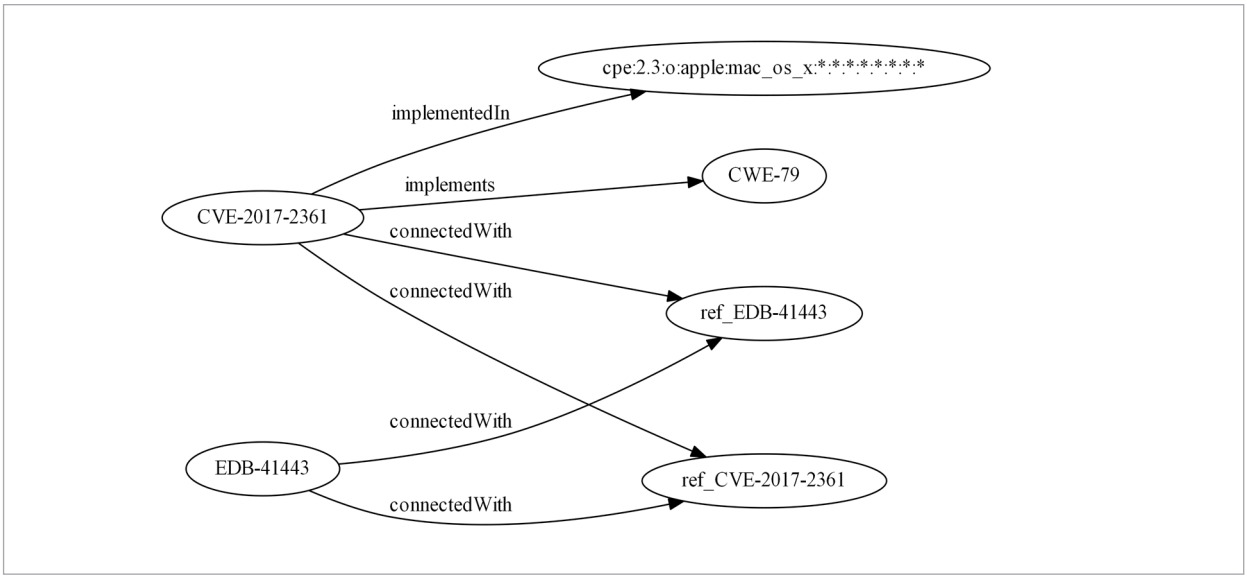
1. "Which weakness(es) is(are) implemented in the vulnerability connected with exploit *EDB-41443?*";
2. "Which product has vulnerability connected with exploit *EDB-41443?*".

The corresponding queries in the DL (Descriptive Logic) Query Language look like this:

1. "*Weakness and implementedBy some (Vulnerability and connectedWith some (Exploit and (localID value "EDB-41443")))*";

Figure 3

An example of sources of security data interrelationships



2 "*Product and containsImplementationOf some (Vulnerability and connectedWith some (Exploit and (localID value "EDB-41443")))*".

The results of these queries are, respectively, the objects: (1) weakness from CWE database "*CWE-79*" and (2) product from CPE database "*cpe:2.3:o:apple:mac_os_x:*:*:*:*:*:*:*:**".

Consideration of additional interconnections on the low level allows one to enhance an accuracy of conclusions. For instance, some CAPEC attack patterns refer to vulnerability instances from the CVE database. This information helps to connect the CVE vulnerability instance with the CAPEC attack pattern instance in more accurate way. However, it requires an additional analysis when this field is filled not for all attack patterns and contains not all possible vulnerabilities.

The conducted experiments and description of the case studies show the existence of opportunity to extend security knowledge through construction and using storage on the basis of the ontological approach.

To generate connections between the ontological concepts, we used explicit references between different security databases. In addition, references on each other, the used databases have implicit connections on the basis of the fields of the same type. For instance, CWE standard has the field "*Applicable Platforms*" for the weaknesses in the CWE database, the CVE vulnerabilities in NVD reference on the platforms in the CPE format, and the attack patterns in the CAPEC database have the field "*Technical Context*".

A comparative analysis of content of these fields allows us to get a more accurate definition of the links between vulnerabilities and attack patterns. Another field that requires additional analysis is related to the attack consequences. Namely, NVD contains values of the CVSS "*Attack Impact*" index for vulnerabilities. This index specifies consequences of vulnerability exploitations. In its turn, the CWE database contains the "*Common Consequences*" field for weaknesses that defines consequences of weakness exploitations.

Finally, the attack patterns in the CAPEC database have the "*CIA Impact*" field. A comparative analysis of content of these fields makes it possible to define a more accurate connection between vulnerabilities and attack patterns.

To form the proposed security information storage, it is needed to use the ontological data model (presented in Section 3) and an access to the sources of security information. It should be noted, that the storage can be updated both by adding new information as well as by modifying old records.

The description of the ontological data model in the OWL language (its type and profile are specified above) allows one to get response in polynomial type ($T(n)=O(n^k)$) which depends on the used resources and the amount of individuals in the storage.

Experimental results show that application of the ontological approach for the security situation analysis can enhance the security management systems. One of its advantages lies in the opportunity to generate a common (not overloaded) data model. This model should be extended for each specific application area.

One of the significant advantages of the ontological approach compared with the relational approach is low resource intensity of the meta scheme modifications. The disadvantage lies in the dependency of quality of the ontological data representation on the quality of the input data.

Thus, the existing security databases frequently contain errors and inaccuracies, and extraction of the interconnections between the objects is obstructed by the absence of the unified reference format, especially for the databases from the different sources.

6. Conclusion

The paper contains the description of the security information storage based on the usage of the ontological and relational approaches. The data model of the proposed storage is based on existing standards for representation of security related data and sources of these data.

To perform the unification of security information from these sources the characteristics and interconnections between them were analyzed. On the basis of these interconnections, the ontological data model for generation of the integrated security data storage was developed and implemented (using Protégé 5.1.0). The proposed data model provides the necessary flexibility to the internal data representation in the repository and the possibility to use the logical inference for more accurate and high-quality queering as well as generates new knowledge based on existing

facts. We believe that this storage and the proposed technique can be a part of the new generation of intelligent systems for security monitoring.

We performed a multitude of experiments to analyze the interconnections between data instances in different security sources and got a statistics on the number of existing links and to prove the possibility to fill the proposed storage. Examples of application of the developed ontology and the technique for derivation of new security related knowledge were demonstrated and discussed.

Future work related to the ontological data model of the proposed security information storage can be divided in following directions: (1) the addition of entities that describe the software and hardware configuration of the protected infrastructure and the analysis of these entities to form links between different types of security information; (2) refinement of the data models for the weaknesses and attack patterns; and (3) the development of the technique for provide the direct and inverse conformity of software and hardware products and their generalized records (on the basis of versions, modifications, revisions, etc.). In addition, we plan to investigate the issues of logical reasoning based on the ontological repository for countermeasure generation and selection, as well as the development of mechanisms for data visualization.

Acknowledgement

This research is supported by the Council for Grants of the President of Russia (project MK-314.2017.9).

References

1. Chechulin, A., Kotenko, I. Attack Tree-Based Approach for Real-Time Security Event Processing. Automatic Control and Computer Sciences, 2015, 49(8), 701-704. https://doi.org/10.3103/S014641161508
2. Common Attack Pattern Enumeration and Classification (CAPEC) Official Website. https://capec.mitre.org. Accessed on January 14, 2018.
3. Common Configuration Enumeration (CCE). https://nvd.nist.gov/cce/index.cfm. Accessed on January 14, 2018.
4. Common Platform Enumeration (CPE) Official Website. https://nvd.nist.gov/cpe.cfm. Accessed on January 14, 2018.
5. Common Remediation Enumeration (CRE) Official Website. https://scap.nist.gov/specifications/cre. Accessed on January 14, 2018.
6. Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org. Accessed on January 14, 2018.
7. Common Vulnerability Scoring System (CVSS) Official Website. https://www.first.org/cvss. Accessed on January 14, 2018.
8. Common Weakness Enumeration (CWE) Official Website. https://cwe.mitre.org. Accessed on January 14, 2018.
9. Doynikova, E., Kotenko, I. Countermeasure Selection Based on the Attack and Service Dependency Graphs for Security Incident Management. In: Lambrinoudakis, C., Gabillon, A. (Eds.), Risks and Security of Internet and Systems, Lecture Notes in Computer Science, 9572, Springer, Cham, 2016, 107-124. https://doi.org/10.1007/978-3-319-31811-0_7
10. Elahi, G., Yu, E., Zannone, N. A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations. In: Laender, A. H. F. et al. (Eds.) Conceptual Modeling – ER 2009. Lecture Notes in Computer Science, 5829. Springer, 2009, 99-114. https://doi.org/10.1007/978-3-642-04840-1_10
11. Fedorchenko, A., Kotenko, I., Chechulin, A. Design of Integrated Vulnerabilities Database for Computer Networks Security Analysis. Proceedings of the 23th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, (PDP 2015), Turku, Finland, March 4-6, 2015, 559-566. https://doi.org/10.1109/PDP.2015.38
12. Fedorchenko, A., Kotenko, I., Doynikova, E., Chechulin, A. The Ontological Approach Application for Construction of the Hybrid Security Repository. Proceedings of the IEEE XX International Conference on Soft Computing and Measurements, Saint-Petersburg, Russia, May 24-26, 2017, 525-528. https://doi.org/10.1109/SCM.2017.7970638
13. Granadillo, G. G., Mustapha, Y. B., Hachem, N., Debar, H. An Ontology-Based Model for SIEM Environments. In: Georgiadis C. K. et al. (Eds.) Global Security, Safety and Sustainability & e-Democracy, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications, 99, Springer, Berlin-Heidelberg, 2012, 148-155. https://doi.org/10.
14. Guo, M., Wang, J. An Ontology-Based Approach to Model Common Vulnerabilities and Exposures in Information Security. http://se.asee.org/proceedings/

ASEE2009/papers/PR2009034GUO.PDF. Accessed on January 14, 2018.

15. Horridge, M. A Practical Guide to Building OWL Ontologies Using Protege 4 and CO-ODE Tools. http://mowl-power.cs.man.ac.uk/ protege
16. IBM X-Force Exchange Project Official Website. http://xforce.iss.net. Accessed on January 14, 2018.
17. ICASI Common Vulnerability Reporting Framework (CVRF) Official Website. http://www.icasi.org/cvrf. Accessed on January 14, 2018.
18. Kotenko, I., Chechulin, A. A Cyber Attack Modeling and Impact Assessment Framework. Proceedings of the 5th International Conference on Cyber Conflict 2013, (CyCon 2013), Tallinn, Estonia, 2013, 119-142.
19. Kotenko, I., Chechulin, A. Computer Attack Modeling and Security Evaluation Based on Attack Graphs. Proceedings of the IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2013), Berlin, Germany, September 12-14, 2013, 614-619. https://doi.org/10.1109/IDAACS.2013.6662998
20. Kotenko, I., Chechulin, A., Doynikova, E., Fedorchenko, A. Ontological Hybrid Storage for Security Data. Proceedings of the 11th International Symposium on Intelligent Distributed Computing, (IDC 2017), Belgrade, Serbia, October 11-13, 2017, 159-171. https://doi.org/10.1007/978-3-319-66379-1_15
21. Kotenko, I., Doynikova, E. Dynamical Calculation of Security Metrics for Countermeasure Selection in Computer Networks. Proceedings of the 24th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, (PDP 2016), Heraklion, Crete, Greece, February 17-19, 2016, 558-565. https://doi.org/10.1109/PDP.2016.96
22. Kotenko, I., Fedorchenko, A., Chechulin, A. Integrated Repository of Security Information for Network Security Evaluation. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2015, 6(2), 41-57. http://doi.org/10.22667/JOWUA.2015.06.31.041.
23. Kotenko, I., Polubelova, O., Chechulin, A., Saenko, I. Design and Implementation of a Hybrid Ontological-Relational Data Repository for SIEM Systems. Future Internet, 2013, 5(3), 355-375. https://doi.org/10.3390/
24. Kotenko, I., Polubelova, O., Saenko, I., Doynikova, E. The Ontology of Metrics for Security Evaluation and Decision Support in SIEM Systems. Proceedings of the 8th International Conference on Availability, Reliability and Security, Regensburg, Germany, September 2-6, 2013, 638-645. https://doi.org/10.1109/ARES.2013.84
25. Metasploit Official Website. https://www.metasploit
26. National Vulnerability Database (NVD) Official Website. https://nvd.nist.gov. Accessed on January 14, 2018.
27. Offensive Security's Exploit Database Archive. https://www.exploit-db.com. Accessed on January 14, 2018.
28. Open Source Vulnerability Database (OSVDB) Blog. https://blog.osvdb.org. Accessed on January 14, 2018.
29. OPENVAS. http://www.openvas.org. Accessed on January 14, 2018.
30. Parmelee, M. C. Toward an Ontology Architecture for Cyber-Security Standards. Proceedings of the 2010 Semantic Technology for Intelligence, Defense, and Security, Fairfax, USA, October 27-28, 2010, 116-123.
31. Protege Wiki Website. Protege User Documentation. https://protegewiki.stanford.
32. PT Application Firewall. https://www.ptsecuri
33. SecurityFocus (BugTraq database) Official Website. http://securityfocus.com. Accessed on January 14, 2018.
34. Sufatrio, Yap, R., Zhong, L. A Machine-Oriented Integrated Vulnerability Database for Automated Vulnerability Detection and Processing. http://citeseerx.ist.psu.edu/viewdoc/
35. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A. UCO: A Unified Cybersecurity Ontology. Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security, Phoenix, Arizona, USA, 2016, 195-202.
36. Tierney, S. Knowledge Discovery in Cyber Vulnerability Databases. https://www.tacoma.
37. Unified Cybersecurity Ontology. https://github.com/Ebiquity/Unified-Cybersecurity-Ontology. Accessed on January 14, 2018.
38. US Computer Emergency Readiness Team (US-CERT). http://www.us-cert.gov. Accessed on January 14, 2018.
39. W3C Website. Web Ontology Language Overview. https://www.w3.org/TR/owl-features. Accessed on January 14, 2018.
40. Waltermire, D., Quinn, S., Scarfone, K., Halbardier, A. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP version 1.2. 2011, 66. https://doi.org/10.6028/NIST.SP.800-126r3
41. Wang, J. A., Guo, M. Security Data Mining in an Ontology for Vulnerability Management. Proceedings of the 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, China, August 3-5, 2009, 597-603. https://doi.org/10.1109/IJCBS.2009.13