# Integrated Repository of Security Information for Network Security Evaluation*

Andrey Fedorchenko, Igor Kotenko†, and Andrey Chechulin
*Laboratory of Computer Security Problems*
*St. Petersburg Institute for Informatics and Automation (SPIIRAS)*
*39, 14 Liniya, St. Petersburg, Russia*
{fedorchenko, ivkote, chechulin}@comsec.spb.ru

**Abstract**

Security evaluation systems usually use various information sources to estimate computer network security. One of the important tasks in these systems is integration and storage of information from various sources. The paper is devoted to investigation and development of models and methods to integrate open security databases into one repository. The model of integration proposed in the paper helps to improve the accuracy of attack detection systems. As sources for security information, different open databases of vulnerabilities, exploits, and dictionaries of products are used, and open databases of weaknesses, attack patterns and configurations are planned to be used. The object of research and development is the mechanisms intended to bind and combine heterogeneous security information. We propose the structure of the integrated repository and the model of security information integration, describe the repository implementation and analyze the results of experiments with the repository.

**Keywords**: security information repository, vulnerability and exploit databases, vulnerability analysis, network security evaluation.

## 1 Introduction

Network security evaluation is one of the very important security problems. In general, network security evaluation can be represented as a complex multi-stage process of detection, verification, comparison and further analysis of threats. The result of this process is an integrated set of metrics that describe the actual security level. A large amount of heterogeneous data from different sources with various formats and semantic content is used for security evaluation. The reason for this is the absence of general agreements, standards and information sources commonly accepted in this field by the world community for selection, specification and analysis of security information. In the best case, some companies or countries have organizations dealing with this issue at the commercial or state levels. Such organizations usually cooperate by creating links to partner's data in its own formats and data. The fact that a large part of security information is closed leads to additional challenge for current research and development in computer security in general.

Beginning from the late of 1980s, open vulnerability databases as well as projects that provide information about released exploits began to appear [2, 3]. At the moment there are several organizations that deal with monitoring, classification and storage of information related to vulnerabilities [4, 5, 6, 7]. These organizations give open access to their vulnerability databases, but each database is filled independently from the others and has its own format of data representation. Besides that, existing vulnerability

*Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, volume: 6, number: 2, pp. 41-57

*This paper is an extended version of the work originally presented at the 23th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'15), Turku, Finland, March, 2015 [1]

†Corresponding author: Tel: +7(812) 328-71-81, Web: `http://www.comsec.spb.ru/`

databases are strictly oriented to the issue how to store, but not how to use data. That leads to significant time consuming to search for vulnerabilities of software and hardware configurations in network security evaluation systems [8, 9, 10, 11, 12, 13, 14, 15, **?**].

Lists of vulnerabilities in these systems are used as main sources for security evaluation, because descriptions of vulnerabilities contain both pre-conditions and post-conditions that characterize the result of possible attacks. Moreover, descriptions of vulnerabilities contain the lists of references to software and hardware products that can be compromised using these vulnerabilities[16].

It is supposed that integration of open security databases (containing vulnerabilities, exploits, attacks, weaknesses, etc.) to one combined database can help to increase the number of unique records that describe vulnerabilities and to extend the list of products that can be attacked using these vulnerabilities. Usage of such database will allow to increase the probability to detect vulnerable software and hardware in computer networks, and, as a consequence, to improve the accuracy of security evaluation.

In addition, implementing the database structure focused on fast search of security information will allow to use this database in the network security evaluation systems, that operate in in near real time[10, 11].

In the paper we explain the main stages needed to design the integrated repository of security information. The aim of the paper is forming the information repository, which includes all required data to evaluate network security.

The paper is an extended version of the work[1] and, besides the design of the integrated vulnerability database, it suggests the relationships between vulnerabilities and exploits records and other security information. These types of data are combined together in an integrated repository of security information.

The novelty of this investigation is in the developed techniques of information integration from different sources and in the proposed approach to the architecture of the repository promoting to increase the speed of accessing to integrated data. In general, it helps to increase the effectiveness of the security evaluation system that uses this repository.

The paper consists of six sections. Second section analyzes related work on systematization and storage of security data. The work devoted to security information sources and integration methods is outlined in third section. Fourth section suggests the architecture of the repository, principles of data integration and schema of access to security data. Fifth section analyzes a prototype of the integrated repository, the results of experiments and usage of the developed repository within the network security evaluation system. Conclusion lists the paper results and plans for the future work.

## 2 Related Work

Vulnerabilities have been detected in a majority of operating systems and applications. As new vulnerabilities are been found permanently, and exploits for them are public, the only way to decrease the probability of their exploitation by attackers is in performing uninterruptable security monitoring, including continuous tracking of appearance of vulnerabilities, operative installation of updates and using instruments that help to counteract to possible attacks, based on these vulnerabilities.

Classifications of vulnerabilities systemize different types of artificial and natural, random and malicious, internal and external threats on many parameters. As a rule, potential security treats are sorted by possibility of exploiting software and hardware vulnerabilities, but classes of vulnerabilities are discussed in existing classification systems only in general. At the same time classifications of vulnerabilities are the basis for creation of threat models for networks security.

Vulnerabilities may be classified by stages of their life cycle, when they appear: design stage, implementation stage, exploitation stage. By object of influence the following types of vulnerabilities are

outlined: of the network level; of the operating systems level; of the database level; of the application level. We also can classify vulnerabilities by the type of the target mean, namely: hardware; operating system; applications.

The first incidents of security breaches, officially registered in vulnerability databases, appeared in 1988. Since then permanent search and registration of vulnerabilities is performed both within different open projects and by commercial companies, research institutes and volunteers. Besides vulnerability databases, there are different exploits databases. Records in these databases describe actions and contain source codes for vulnerability exploitation. One can find an exploit for a specific vulnerability by identifiers of vulnerability databases, or by names of vulnerable hardware and software.

At present there is a large number of companies, projects, institutes and government organizations that specifically perform search of vulnerabilities in software and hardware for many years. Their vulnerability databases contain enormous number of records. Databases of vulnerabilities of product manufacturers are less by the number of vulnerabilities and local distribution, but not less by value.

Every year the growth of detected vulnerabilities along with the increase of products are kept on high level. Every vulnerability database has its own description structure, and though most often similar items and vulnerability representations are met, there are also significant differences between databases. The systematization task is complicated by the lack of a common description format. The estimation of network security by means of multitude of vulnerability databases may become hard to fulfill already in several years.

Among the leaders of vulnerability detection, we may outline:

- MITRE and its Common Vulnerabilities and Exposures (CVE) [4];
- National Institute of Standards and Technology (NIST) and its National Vulnerabilities Database (NVD) [5];
- The project Open Source Vulnerabilities Data Base (OSVDB)[6];
- United State Computer Emergency Readiness Team (US-CERT) with Vulnerability Notes Database (VND) [17];
- The project SecurityFocus with its vulnerability database BugTraq [18];
- IBM with vulnerability database X-Force [11];
- As well as commercial databases of companies Secunia [19] and VUPEN Security [20], etc.

To integrate vulnerability databases there were already suggested methods of integration of information from different sources [15, 21]. These papers are based on automated identification of records by their identifiers, but without consideration of product records, that is an essential part of every vulnerability database.

The models and methods for integration of vulnerability databases are proposed for using in automated systems of security evaluation in a number of papers [15, 22]. In [22] the classifications of vulnerabilities based on errors classes, authentication necessity and impact levels are investigated. This paper also contains the model of vulnerability databases classifications based on the level of maintenance.

In [8, 9, 12] the approaches are suggested that are based on ontologies to represent and manipulate vulnerabilities. It is supposed that these approaches may be used for classification and administration of vulnerabilities represented in the NVD database [5].

The paper [23] is aimed to design and implement a service for collection of vulnerability information. This approach has the following distinction from many others: it uses the original schema for adding new

vulnerability records from open databases as well as by users. The architecture of the service is based on the vulnerability classification by types and functional groups.

The paper [14] is dedicated to the project performed in MITRE on creation of the general ontology for Security Content Automation Protocol (SCAP) [14]. In the most general form SCAP acts as follows:

- based on Common Platform Enumeration (CPE) the list of the products is created (and is used for management of actives [24];
- taking into account Common Configuration Enumeration (CCE) the list of configuration features of these products, that negatively affect on security, is created (used for management of configurations) [25];
- based on Common Vulnerabilities and Exposures (CVE) the list of vulnerabilities of these products is created (used for vulnerability management) [4];
- Common Vulnerabilities Scoring System (CVSS) is used for estimation of negative effect of configurations and vulnerabilities, detection of the most critical vulnerabilities [2].

The work on SCAP is not finished yet.

In [3, 26] the integration of vulnerability databases based on the NVD database is considered. The technique of integration, suggested in [26], uses CVE identifiers [4] as the main feature. Implementation of the proposed technique uses the relational database MySQL which is installed in FreeBSD.

There are also commercial products, which are based on integration of vulnerability records from public vulnerability databases and vendors databases, namely SAINTscanner [27] and McAfee Vulnerability Manager for Databases [28].

There are also a number of exploit databases. Examples of the leading ones are Exploit Data Base (EDB) [29] and the 1337day project [30]. The second one, in contrast to the first, contains some exploits for sale. It also contains information about products (including their versions) for which exploits are applicable.

Metasploit framework [31] occupies a special place among tools that search and apply exploits. This product is designed for penetration testing. It provides opportunity to search exploits by the identifiers of vulnerabilities from the leading databases or by the targeted software and hardware.

Aspects of using vulnerability information for attack graph construction and vulnerability analysis are considered, for example, in [32, 33, 34].

Analysis of related work shows that at present the integration of ontological and relational approaches are actively developed for representation and management of vulnerabilities.

The advantages of these approaches are possibility to create a data model in a most general and at the same time not overloaded way, that must be adapted for every application domain in the process of deployment. This feature is especially important for security systems that are being deployed in very different domains, including critical infrastructures.

However we cannot also avoid mentioning disadvantages of this approach, including, for example, increase of time for vulnerability search compared to the specialized relational databases [13].

In this paper, based on existing standards, we design relational models of different types data sources and structure them in the common repository. Results are used within the frames of software, intended for security information and events' management. We plan to use a relational DBMS as data storage for vulnerabilities.

## 3 Security information sources

The security information sources, that are proposed to be used in the final structure of the integrated repository, have different types, namely:

1. Vulnerabilities.
2. Hardware and software.
3. Exploits (with source files or text files of implementation details description).
4. Weaknesses.
5. Optimal configurations (for security).
6. Attack patterns.

It has to be also noticed that there are some additional data types, that can be useful for security evaluation:

1. Remediations (after destructive attack actions).
2. Updates (up to hardware and software patches that are not defined in products dictionaries).
3. Signatures of malware, scripts and shell codes.

In the paper, vulnerabilities, software/hardware products and exploits were chosen as the main types of security information. These types are independent units of the integrated security repository. This is because each of them can help to define the security level or a property of the analyzed network. In turn, each type directly related with two other types. However, this relationship can be both direct and indirect. For instance, each vulnerability refers to at least one product name. In turn, each exploit refers to at least one vulnerability. Hence, at least one product name record exists for each exploit record. In the first two cases the relationships are direct. In third case it is indirect and provided by analysis of identifiers of vulnerabilities and exploits. However, it is also possible to set the correspondence between records of these types by referencing in vulnerability records to exploit records.

The vulnerability database CVE [4] is maintained since 1999 and as of April 10, 2015 contained 80 094 records. The main difference of this database is that it is the most complete and systemized. Thus, it is used as a basis for noting correspondence of vulnerability records in other databases. The vulnerability database CVE contains a sufficiently complete vulnerability list and has a big number of links to vulnerability databases and software/hardware producer databases (Fig. 1).

On the other side, the CVE database lacks a mechanism for describing the relation of vulnerabilities to concrete products, as well as assigning them metrics and calculation of impact values. This drawback is resolved by expanding the list of vulnerability properties by the data from the NVD database [5]. This database also includes vulnerability descriptions along with links to corresponding vulnerable hardware and software in the CPE format [24]. Besides, this database contains indices, characterizing vulnerabilities in the CVSS format [35, 2]. Analysis of the NVD database showed that the relations include only 10.63% of records of all vulnerabilities. These indices gave possibility to use non-separable records (in one table) of product configurations to the integrated vulnerability database according to the designed structure, without making long-time calculations. OSVDB is an independent and open vulnerability database, created for community of specialists in security area. The aim of the OSVDB creation project is to provide accurate, detailed, actual information on vulnerabilities for security systems [6]. Analyzing

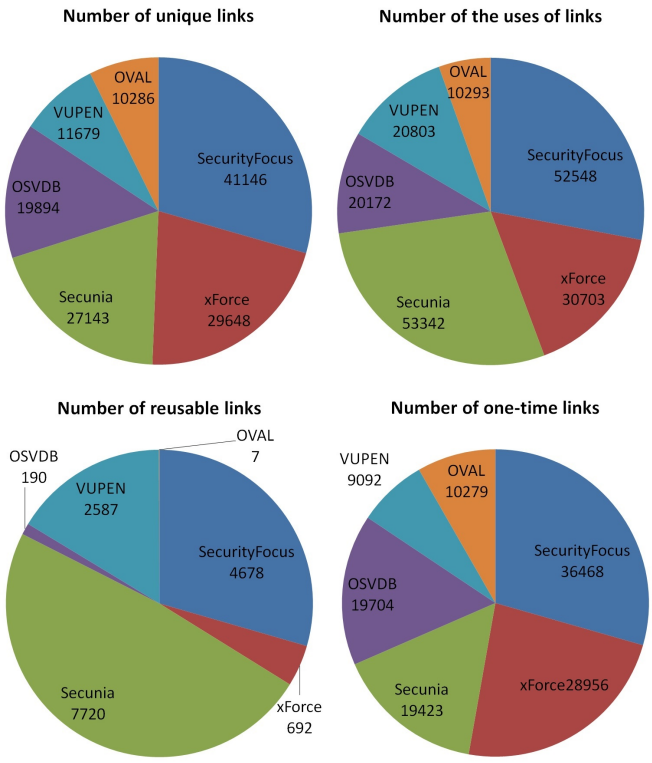



Figure 1: Statistics of presence of links to external sources and vulnerability descriptions for the CVE database

the OSVDB database, a lot of links to external sources were detected, and the best indices on quality and uniqueness are links to the CVE and NVD databases (Fig. 2).

The integrated vulnerability database also was suggested to be extended by the X-Force database [7]. The X-Force vulnerability database is the project of IBM and is in open access in the Internet. It should be noted that the X-Force database, besides basic characteristics of vulnerabilities in the CVSS format, has temporal indices, being unique among the vulnerability databases used.

Based on all facts specified above, these databases became the main sources of information about vulnerabilities for forming the integrated vulnerability database.

The Common Platform Enumeration (CPE) [24] was taken as a basis for the integrated products dictionary. In addition, the format of CPE v.2.2 was taken to specify product name records. This format was chosen due to precise and systematized representation of product format, which contains product type, vendor, title, update and edition.

The Exploit Data Base (EDB) [29] is an archive of exploits and vulnerable software. The goal of this project is to collect exploits from different sources (private authors, e-mail lists, etc.) and combine them in an unified database with easy navigation. The following fields are included in the EDB record structure: EDB identifier; exploit name; platform; exploit type; publication date; author; CVE identifier; OSVDB identifier; accept flag. In addition, each record in this database contains a source file (or text description of instructions and details of implementation) for vulnerability exploitation. An installation packet or an executable file of vulnerable product is also available for some exploit records. It has to be noticed that the fields of CVE and OSVDB identifiers are optional.

The extra attention should be paid to the 1337day.com resource, where exploits are openly sold. This resource is one of the fullest sources of exploit records. At the same time, this database is just

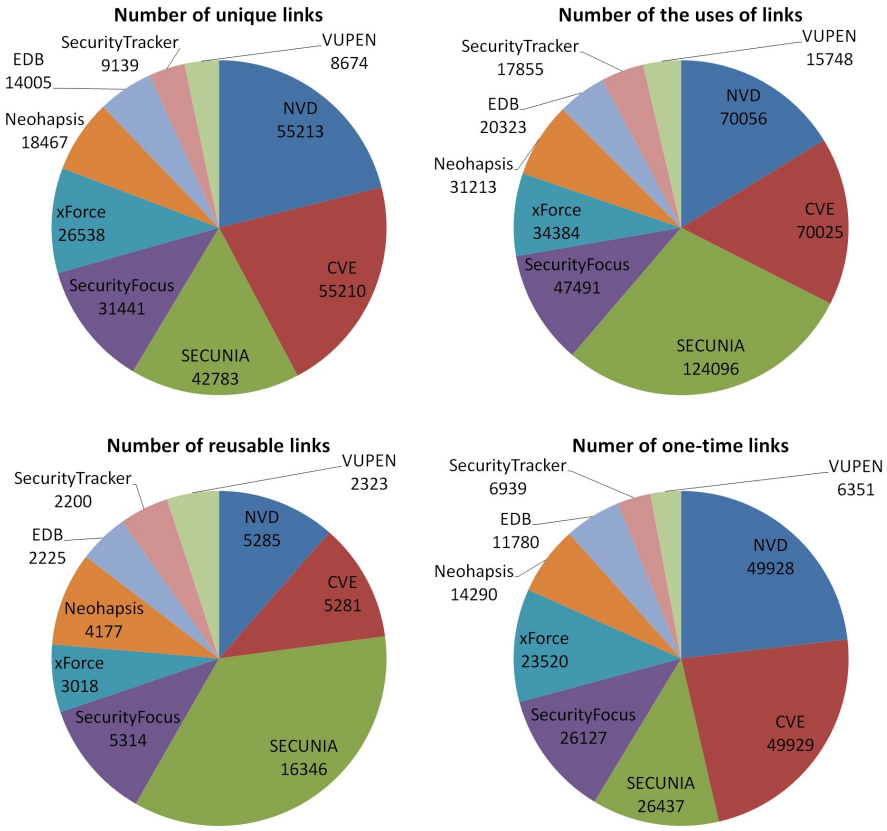



Figure 2: Statistics of presence of links to external sources of vulnerability descriptions for the OSVDB database

slightly different from EDB according the number of public exploits. However, this resource may play an important role in security analysis due to existence of paid and ”closed” exploits. For example, when new exploits for a specific product appear on market, it is possible to monitor this case in on-line mode. Existence of such software in the evaluated network can lead to decrease of security level. This fact can be also used in security event correlation. For instance, it should be taken into account, if a software with a corresponding exploit has anomalies in behavior.

# 4 Integrated Repository Architecture

Design of the integrated repository of security information (IRSI), specified in the paper, has in mind both the integration of data of different types and the integration of data of one type from different sources. For example, in the IRSI architecture we suggest to include the integrated vulnerability database. Currently this database consists of several open databases. Every such database has its own format of vulnerability description.

Thus, to design the IRSI it is necessary to:

1. Make an integration of data structures of one information type in the scope of each component unit of the repository.

2. Define a relationship between data of different types.

3. Develop the common model of data integration in the repository on the base of two previous stages.

In common view, at this stage, the IRSI architecture looks as follows. On the level of data types the records of vulnerabilities, exploits and products are disposed. These types form the integrated vulnerability database, integrated exploits database and integrated products dictionary. We call them as IRSI subrepositories (or components).

External sources of security information compose the internal architecture of each subrepository (Fig. 3).

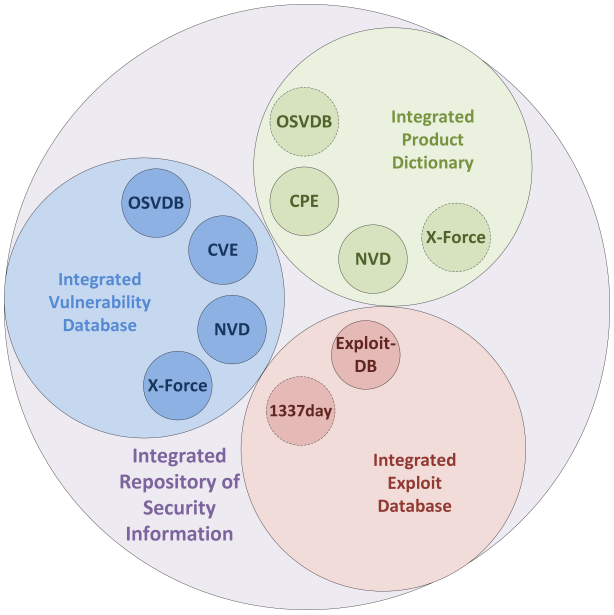



Figure 3: Common architecture of IRSI

Fig. 3 shows that some sources (for example, NVD, OSVDB or X-Force) are used in components of different types. This fact is determined by existence in the specified sources the records on vulnerabilities and the records on names of products, which vulnerabilities may be exploited. It should be noted, that the CPE dictionary includes not all entries of software and hardware. Therefore this source cannot be a single resource to form the integrated product dictionary. The sources that are marked by dotted lines, are included in corresponding subrepository conditionally. For example, some sources of product names do not have the common format of name specification. In this case, complexity of using them both for forming and for searching products increases. However, they could be used. In case of using the 1337day resource in the Integrated Exploit Database (IED), this conditional character consists in detection of and binding with only exploits having paid access.

Fig. 4 depicts the connections between sources in use. Such view allows to understand on what principles the model of the integrated repository generation should be based. For example, it is clear, that the NVD database is related with remaining security data sources as used source (an incoming arrow) and as source that uses other sources (an output arrow). The identifier of the CVE database is selected as the basis of the integrated vulnerability database. It results from the fact that the CVE and NVD databases use the same identifiers and these databases are the most complete catalogues of vulnerabilities. In Fig. 4, the source of the CVE database is designated as CVRF (Common Vulnerability Reporting Framework) [36]. This is due to the fact that currently the description of vulnerabilities in CVE is realized only in the specified format.

Dotted lines in Fig. 4 designate the relationships between sources, which records can be mapped only

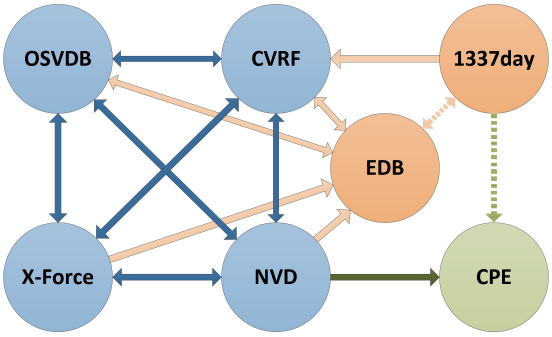



Figure 4: The schema of relationships between used sources of security information

after an additional information transformation. For example, records of the 1337day resource indicate vendor, name and version of the product applicable for the exploit. But this representation form does not allow to carry out mapping of product records of this resource into the CPE dictionary without additional name recognition operations.

The advantage of the described IRSI architecture is possibility of access to its data by using the schema depicted in Fig. 5. In Fig. 5, the configurations of software and hardware are used as an input to get security information. Then, the internal identifier (IPD_ID) of the integrated product dictionary (IPD) is defined on the base of each product record of used configurations. As a result, by values of mentioned identifiers, it is possible to determine the existence of vulnerability records in the integrated vulnerability database (IVDB), where parameter IVDB_ID is the identifier of every such record. The next stage includes output of the information about the found vulnerability, or using the parameter IEDB_ID to find available exploits in the integrated exploit database (IEDB). Thus, an output includes description of found records of vulnerabilities and exploits, that are used directly in the security evaluation system.

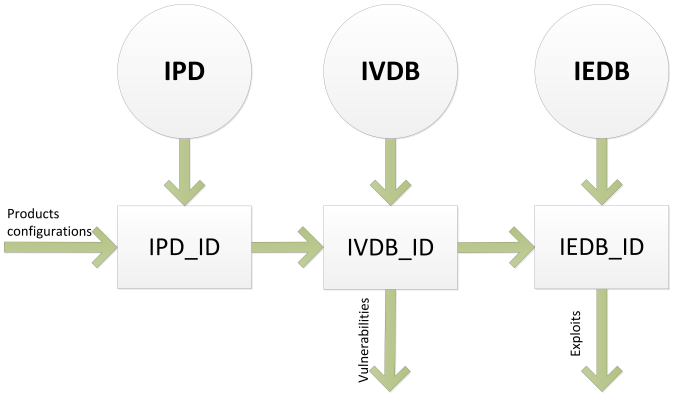



Figure 5: Schema of access to IRSI data

## 5 Integrated repository prototype: experiments and efficiency analysis

Based on the proposed IRSI architecture, an IRSI prototype was developed. It is intended for practical estimation of the IRSI efficiency and validation of its application in the network security evaluation system.

The prototype is designed to solve the following two main practical problems:

1. Generation of the integrated repository.
2. Its exploitation as a component within the networks security evaluation system on given input data.

Fig. 6 outlines the proposed architecture of the IRSI prototype.
The IRSI prototype includes following modules:

- Load Manager — loads the CVE vulnerability list, vulnerability databases, the CPE dictionary, and exploits;
- Composite Manager — integrates information about products, vulnerabilities and exploits;
- DataBase Manager — creates, cleans and interacts with IRSI;
- Web Service — performs remote network access to the operations of monitoring vulnerabilities of the network hosts. In addition, this module performs monitoring of new exploits published on 1337day.com.

To add other sources of security information, it is necessary to add the corresponding processing functions to the Load Manager module, without any influence to the work of other modules.

This facility and modular architecture, in a whole, proves the possibility of flexible modification of the designed prototype to increase the number of used databases as well as used data types (weaknesses, attack templates, etc.).

Let us consider statistics on vulnerability integration processes. To form the integrated vulnerability database the following sources of vulnerabilities and product descriptions were used: CVE database; NVD database; OSVDB database; X-Force database; CPE dictionary; Exploit-DB database; 1337day resource.

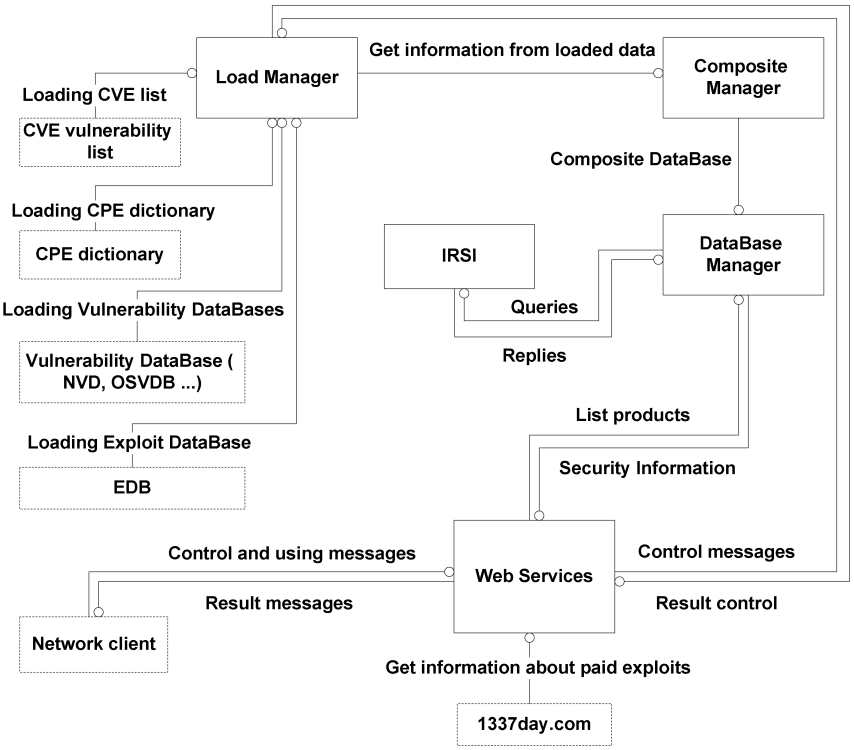



Figure 6: The architecture of the IRSI prototype

Fig. 7 shows the scheme of IRSI generation. This scheme is implemented in LoadManager and CompositeManager modules. In the scheme it is assumed that the CPE product dictionary is already loaded in the general products space, and unique identifiers are assigned to all records. It should be noticed that the general list of products references includes links to records from other sources of security information.

The scheme is divided into two parts.

First one shows the loading of source data from various sources related to information security. In this part each record from the external database is processed and converted to the internal format. During this process the general list of vulnerabilities and products is generated.

Second part is devoted to the IRSI construction. Firstly each record is matched with others by the links to other databases. Then the references to this record from other databases are analyzed. On the next step all connected records are joined to one record. It should be noticed that the scheme for vulnerability mapping is suitable for information related to exploits as well. The final action in the scheme is the direct recording of grouped records with internal identifiers into new integrated repository of security information.

As a result of forming the integrated database the total number of unique vulnerability records in comparison with other used sources increased at least by 30%, and the number of the records of the integrated product dictionary—by 110%.

Quantitative indices of the number of unique vulnerability records for the used vulnerability databases and the generated IVDB are considered in Table 1.

Table 1: The number of unique vulnerability records in different databasess

| Database of vulnerabilities | Number of unique vulnerability records |
|---|---|
| CVE | 45270 |
| OSVDB | 58570 |
| X-Force | 54128 |
| IVDB | 85171 |

Taking into account Table I, we may come to conclusion on the evident superiority of the IVDB on the number of unique records over the CVE, OSVDB and X-Force databases used as sources of information about vulnerabilities.

During the analysis of the EDB, it was found that 60% of all exploit records are linked with vulnerabilities from the CVE database and 86% of them are unique (Fig. 8).

In addition, it has to be noticed that only 25% records of the CVE vulnerabilities have public exploits in EDB (Fig. 9).

However, besides the large amount of vulnerability records with public exploits, nowadays the market of zero-day vulnerabilities and exploits is actively growing. Many of them will never be shared.

Considering the obtained quantitative indices of unique vulnerability records as well as the calculated indices of unique products records, we performed the comparative analysis of the integrated products list and the integrated products dictionary related to described data sources. Results of the analysis are presented in Fig. 10. Values in the table show percentages occupied by data of the corresponding database in the IVDB.

The presented prototype of the integrated database was used as the base component to create the automated system for attack modeling and security evaluation [10, 11].

A set of experiments was held to evaluate the IRSI efficiency for modeling of different scenarios of attacks. All experiments were fulfilled on the simulation testbed with the following characteristics:

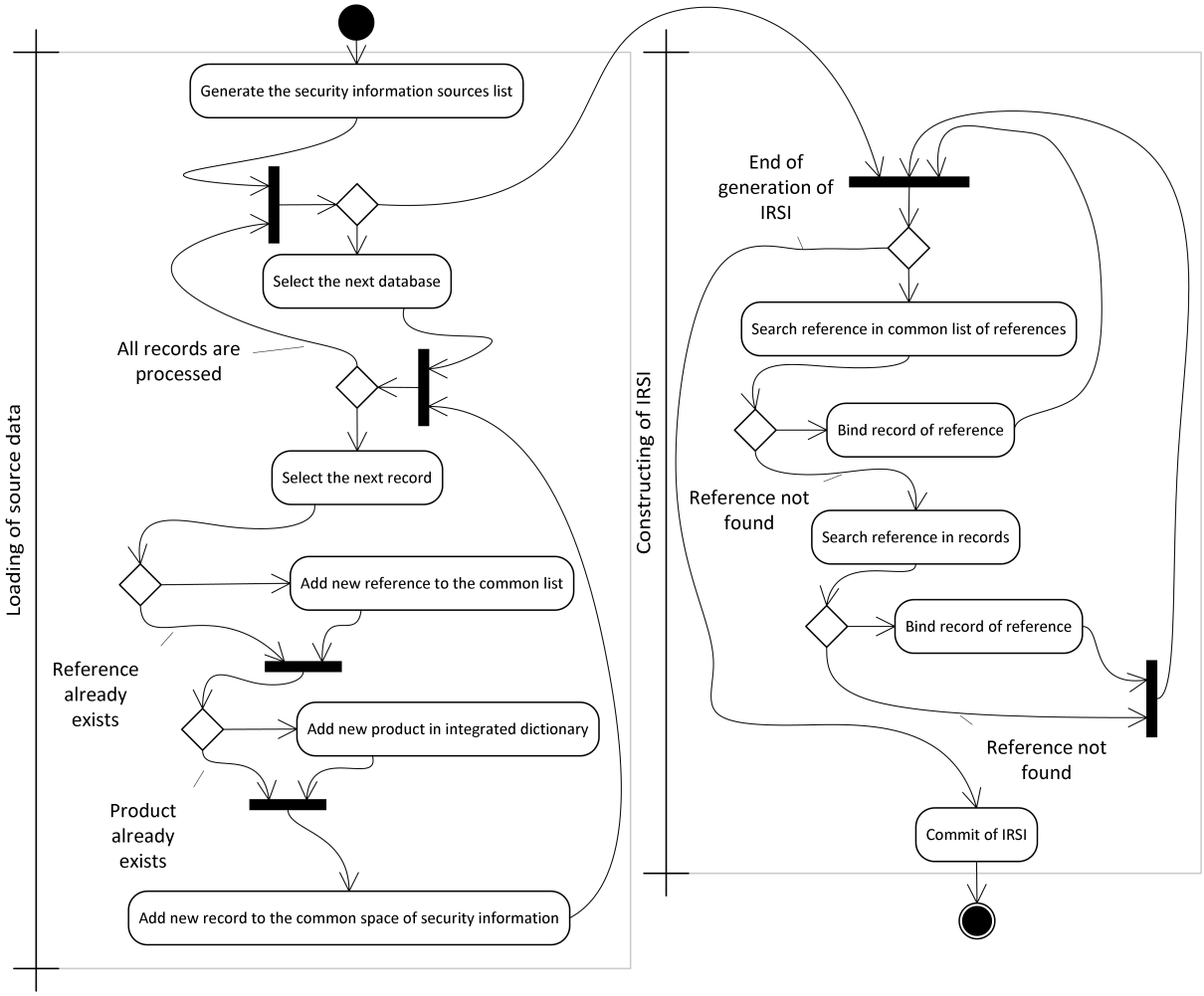



Figure 7: IRSI generation scheme

processor Intel Core i5-3570 3,4 GHz (2 core, 4 thread); RAM — 16 GB, 1333 MHz; HDD — 500 GB; operating system — Microsoft Windows7.

The designed prototype was implemented in Java using JDK v.7. The prototype work management was performed by application server Apache Tomcat v.6. In the course of experiments two modes of work of the prototype were outlined: (1) IRSI generation; (2) using IRSI as a component of the network security evaluation system.

The experiments resulted in the indices of required resources and time in the specified work modes (Table 2).

The presented results were calculated using software instruments built for analysis of used resources and outline the maximal values of the series of performed experiments. Thus, the time of the complete repository generation is not more than 35 minutes, and the maximal temporal indices for requests processing and results forming for one host of the evaluated network are 0.063 seconds.

Based on the performed estimation of the attack graph modeling and security evaluation component [10, 12], in which there was set the affordable time for forming results about vulnerabilities of known beforehand products configuration (equal to 1 minute for 1000 hosts of a random network), we conclude that the designed prototype meets the requirements on operativeness for further usage within the security evaluation system in the near real time mode.

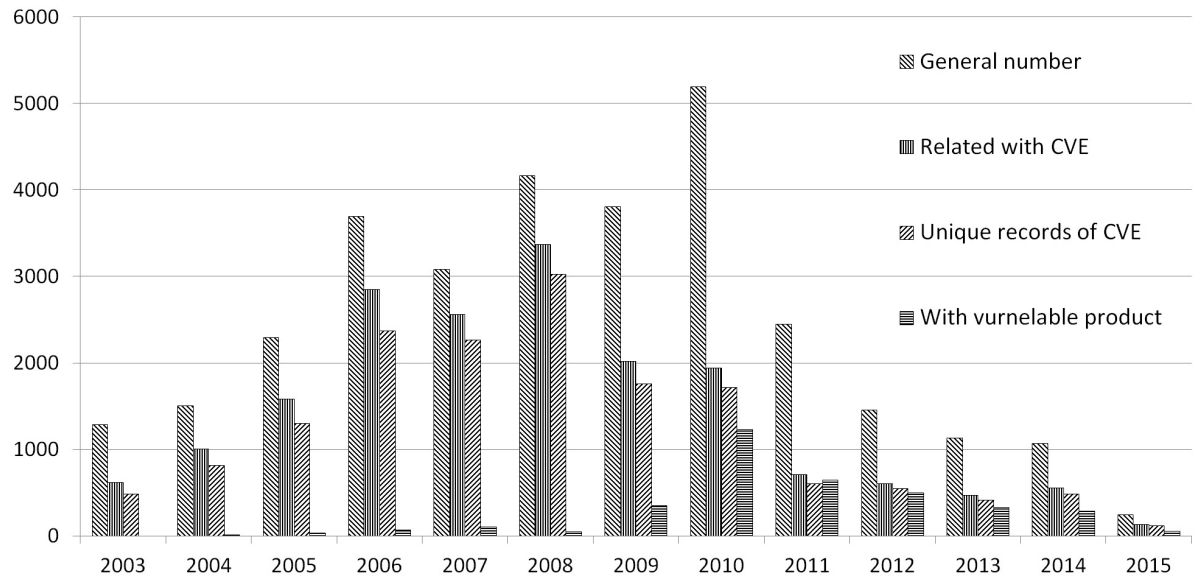



Figure 8: Statistics of relationship between EDB and CVE database records in 2003-2015

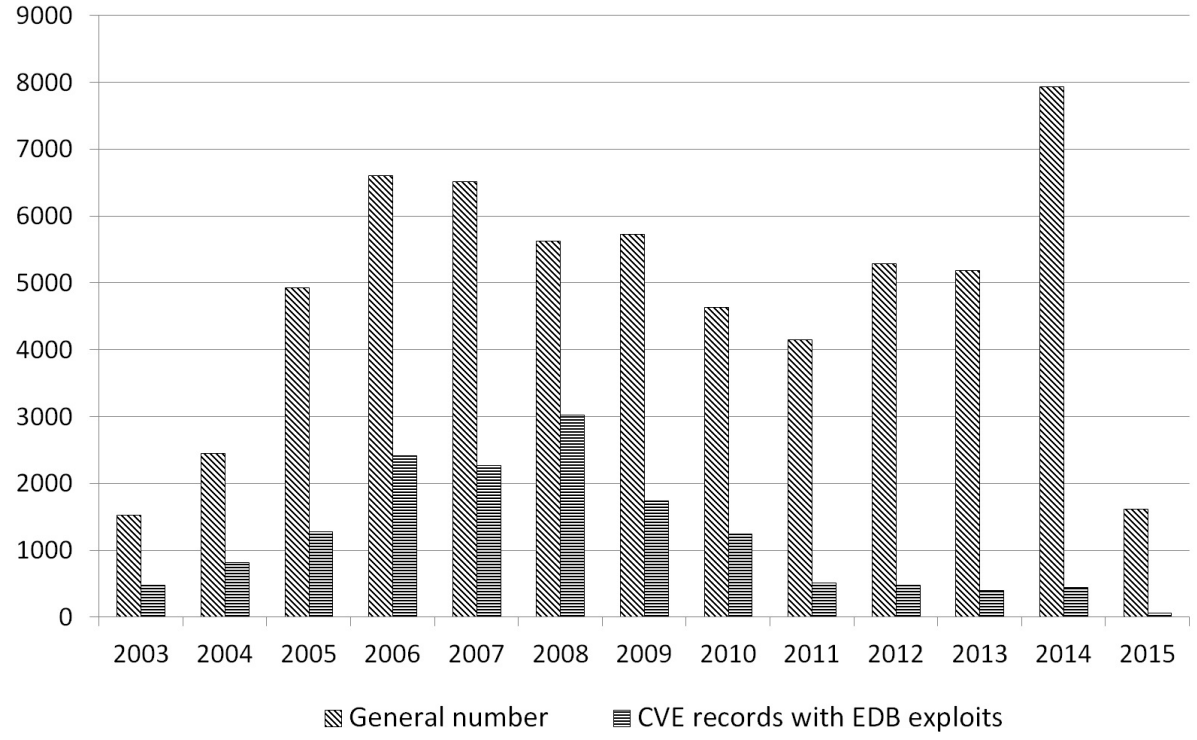



Figure 9: EDB records in the CVE database in 2003-2015

The results of experiments have confirmed efficiency of the designed security information repository within the security evaluation system.

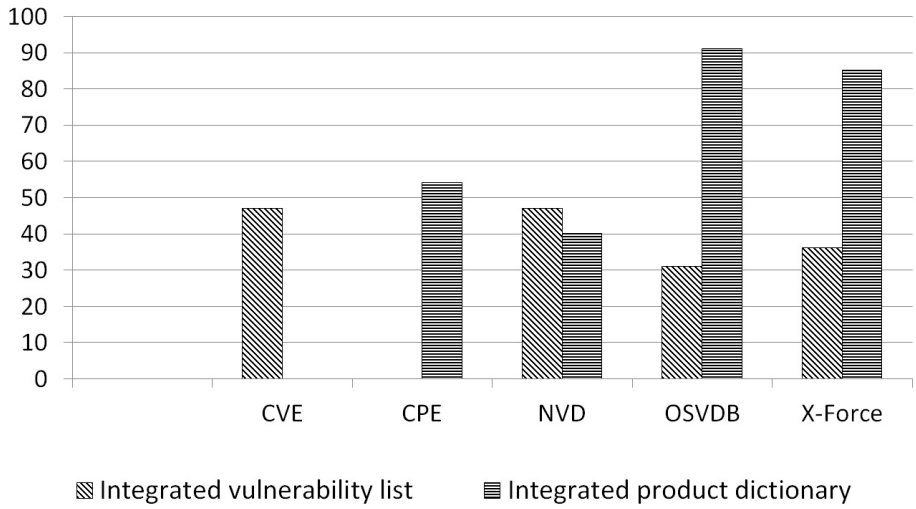



Figure 10: Results of comparative analysis of data sources

Table 2: Indices of required resources

| Used resource | Indices (max. values) | |
|---|---|---|
| | Generation of IRSI | Using IRSI within the network security evaluation system |
| Processor load(%) | 31 | 12 |
| RAM (MB) | 1800 | 250 |
| Physical memory (MB) | 1700 | 1700 |
| Execution time | 35 minutes | 0,063 sec. |

# 6 Conclusion

The paper proposed the architecture and main principles of integrated security information repository. It can be used by security evaluation system for computer infrastructures and intended for operating in near real-time.

The developed integrated repository of security information helps to increase the detection accuracy of network security weaknesses and find the existing public exploits for them. This functionality allows to increase efficiency of network security evaluation.

In the future work we plan to upgrade the structure of integrated repository of security information by adding in integration process other data types, namely: hardware and software weaknesses and attack patterns.

For the integrated product dictionary we currently use the CPE dictionary of version 2.3 as a basis and plan to use the dictionary in the Common Vulnerability Reporting Framework (CVRF) format [36]. The second one by its structural characteristics will allow us to avoid ambiguities in record names. Also we plan to enhance the update function to exclude the necessity of complete reforming of the integrated repository of security information. This will allow us to increase the efficiency of the repository generation.

## Acknowledgement

This research is being supported of The Ministry of Education and Science of The Russian Federation (contract # 14.604.21.0033, unique contract identifier RFMEFI60414X0033).

## References

[1] A. Fedorchenko, I. Kotenko, and A. Chechulin, "Design of integrated vulnerabilities database for computer networks security analysis," in *Proc. of the 23th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'15), Turku, Finland*. IEEE, March 2015, pp. 559–566.

[2] K. Scarfone and P. Mell, ""an analysis of cvss version 2 vulnerability scoring"," in *October*. IEEE, Proc. of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09), Lake Buena Vista, Florida, USA 2009, pp. 516–525.

[3] G. Kim, J. Oh, D. Seo, and J. Kim, "The Design of Vulnerability Management System," *International Journal of Computer Science and Network Security*, vol. 13, no. 4, April 2013.

[4] "Common Vulnerabilities and Exposures (CVE)," http://cve.mitre.org, [Online; accessed April-2015].

[5] "National Vulnerabilities Database (NVD)," http://nvd.nist.gov, [Online; accessed April-2015].

[6] "Open Source Vulnerabilities Data Base (OSVDB)," http://osvdb.org, [Online; accessed April-2015].

[7] "X-Force," http://xforce.iss.net, [Online; accessed April-2015].

[8] G. Elahi, E. Yu, and N. Zannone, "A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations," in *Proc. of the 28th International Conference on Conceptual Modeling (ER'09), Gramado, Brazil, LNCS*, vol. 5829. Springer Berlin Heidelberg, November 2009, pp. 99–114.

[9] M. Guo and J. Wang, "An Ontology-based Approach to Model Common Vulnerabilities and Exposures in Information Security," in *Proc. of the 2009 ASEE Southeast Section Conference, Marietta, Georgia, USA*. American Society for Engineering Education, April 2009.

[10] I. Kotenko and A. Chechulin, "A Cyber Attack Modeling and Impact Assessment Framework," in *Proc. of the 5th International Conference on Cyber Conflict (CyCon'13), Tallinn, Estonia*. IEEE, June 2013, pp. 119–142.

[11] ——, "Common Framework for Attack Modeling and Security Evaluation in SIEM Systems," in *Proc. of the 2012 IEEE International Conference on Green Computing and Communications (GreenCom'12), Besançon, France*. IEEE, November 2012, pp. 94–101.

[12] I. Kotenko, I. Saenko, O. Polubelova, and A. Chechulin, "Design and Implementation of a Hybrid Ontological-Relational Data Repository for SIEM systems," *Future internet*, vol. 5, no. 3, pp. 355–375, 2013.

[13] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: are they so different? A comparison," *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, December 2012.

[14] M. C. Parmele, "Toward an Ontology Architecture for Cyber-Security Standards," in *Proc. of the 5th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS'10),Fairfax, Virginia, USA, CEUR*, vol. 713, October 2010, pp. 116–123.

[15] Sufatrio, R. Yap, and L. Zhong, "A Machine-Oriented Integrated Vulnerability Database for Automated Vulnerability Detection and Processing," in *Proc. of the 18th Large Installation System Administration Conference (LISA'04), Atlanta, Georgia, USA*. USENIX Association, November 2004, pp. 47–58.

[16] J. Ruiz, R. Harjani, A. Maña, V. Desnitsky, I. Kotenko, and A. A. Chechulin, "Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components," in *Proc. of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'12), Garching, Germany*. IEEE, February 2012, pp. 261–268.

[17] "United States Computer Emergency Readiness Team (US-CERT)," http://www.us-cert.gov, [Online; accessed April-2015].

[18] "BugTraq," http://securityfocus.com/, [Online; accessed April-2015].

[19] "Secunia," http://secunia.com, [Online; accessed April-2015].

[20] "Vupen Security," http://www.vupen.com, [Online; accessed April-2015].

[21] S. Tierney, "Knowledge Discovery in Cyber Vulnerability Databases," 2005.

[22] M. Schumacher, C. Haul, M. Hurler, and A. Buchmann, "Data Mining in Vulnerability Databases," in *Proc. of the 7th Workshop "Sicherheit invernetzten Systemen", Hamburg, Germany*, February 2000.

[23] L. Ma, S. Mandujano, G. Song, and P. Meunier, "Sharing Vulnerability Information using a Taxonomically-correct, Web-based Cooperative Database," CERIAS, Tech. Rep. 2001-03, February 2001.

[24] "Common Platform Enumeration (CPE)," http://cpe.mitre.org, [Online; accessed April-2015].

[25] "Common Configuration Enumeration CCE," http://cce.mitre.org, [Online; accessed April-2015].

[26] P. Huang and C. Yang, "Research and Integration of Heterogeneous Vulnerability Database," in *Proc. of the 2009 Computational Intelligence Signal Processing (CISP'09), Taipei, Taiwan*, April 2009.

[27] "SAINTscanner," http://www.saintcorporation.com/solutions/vulnerabilityScan.html, [Online; accessed April-2015].

[28] "McAfee Vulnerability Manager for Databases," http://www.mcafee.com/uk/products/vulnerability-manager-databases.aspx, [Online; accessed April-2015].

[29] "Exploit Data Base," http://exploit-db.com, [Online; accessed April-2015].

[30] "1337Day Inj3ct0r Exploits Market and 0day Exploits Database," http://1337day.com, [Online; accessed April-2015].

[31] "Metasploit," http://www.metasploit.com, [Online; accessed April-2015].

[32] S. Roschke, F. Cheng, R. Schuppenies, and C. Meinel, "Towards unifying vulnerability information for attack graph construction," in *Proc. of the 12th International Conference on Information security (ISC'09), Pisa, Italy, LNCS*, vol. 5735. Springer Berlin Heidelberg, September 2009, pp. 218–233.

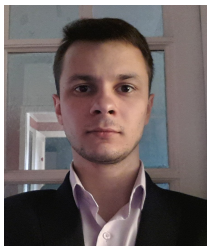
[33] S. Roschke, F. Cheng, and C. Meinel, "Using Vulnerability Information and Attack Graphs for Intrusion Detection," in *Proc. ot the 6th International Conference on Information Assurance and Security (IAS'10), Atlanta, Georgia, USA*. IEEE, August 2010, pp. 104–109.

[34] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in *Proc. of the 2006 SIGCOMM workshop on Large-scale Attack Defense (LSAD'06), Pisa, Italy*. ACM, September 2006, pp. 131–138.

[35] "Common Vulnerability Scoring System (CVSS)," http://www.first.org/cvss, [Online; accessed April-2015].

[36] "Common Vulnerability Reporting Framework (CVRF)," http://www.icasi.org/cvrf-1.1, [Online; accessed April-2015].

## Author Biography



**Andrey Fedorchenko** graduated from St. Petersburg State Electrotechnical University. He is a PhD student at St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) and holds a position of junior researcher at the Laboratory of Computer Security Problems of SPIIRAS. He has a experience in the research on computer network security and participated in several projects on developing new security technologies. His primary research interests include computer network security, intrusion detection and malware analysis.

**Igor Kotenko** graduated with honors from St.Petersburg Academy of Space Engineering and St. Petersburg Signal Academy. He obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Head of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. He is the author of more than 250 refereed publications, including 12 textbooks and monographs. Igor Kotenko has a high experience in the research on computer network security and participated in several projects on developing new security technologies. For example, he was a project leader in the research projects from the US Air Force research department, via its EOARD (European Office of Aerospace Research and Development) branch, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. The research results of Igor Kotenko were tested and implemented in more than fifty Russian research and development projects.

**Andrey Chechulin** received his B.S. and M.S. in Computer science and computer facilities from Saint-Petersburg State Polytechnical University and PhD from St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) in 2013. In 2015 he was awarded the medal of the Russian Academy of Science in area of computer science, computer engineering and automation. At the moment he holds a position of senior researcher at the Laboratory of Computer Security Problems of SPIIRAS. He is the author of more than 50 refereed publications and has a high experience in the research on computer network security and participated as an investigator in several projects on developing new security technologies. His primary research interests include computer network security, intrusion detection, analysis of the network traffic and vulnerabilities.