2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing

Common Framework for Attack Modeling and Security Evaluation in SIEM Systems

Igor Kotenko and Andrey Chechulin
Laboratory of Computer Security Problems
St. Petersburg Institute for Informatics and Automation (SPIIRAS)
Saint-Petersburg, Russia
{ivkote, chechulin}@comsec.spb.ru

***Abstract*—The paper suggests a framework for attack modeling and security evaluation in Security Information and Event Management (SIEM) systems. It is supposed that the common approach to attack modeling and security evaluation is based on modeling of a malefactor's behavior, generating a common attack graph, calculating different security metrics and providing risk analysis procedures. Key elements of suggested architectural solutions for attack modeling and security evaluation are using a comprehensive security repository, effective attack graph (tree) generation techniques, taking into account known and new attacks based on zero-day vulnerabilities, stochastic analytical modeling, and interactive decision support to choose preferred security solutions. The architecture of the Attack Modeling and Security Evaluation Component (AMSEC) is proposed, its interaction with other SIEM components is described. We present the prototype of the component and the results of experiments carried out.**

Keywords-Attack modeling; Security evaluation; SIEM; Attack graph; Service dependences; Zero day vulnerabilities

I. INTRODUCTION

The complexity of computer network security management causes the necessity to develop powerful automated security analysis components which can be important components of Security Information and Event Management (SIEM) systems [21, 23]. These components should allow finding and correcting errors in the network configuration, reveal possible assault actions for different security threats, determine critical network resources and choose an effective security policy and security mechanisms appropriate to current threats.

The paper considers attack modeling security evaluation processes, intended to be implemented for the security analysis in SIEM systems. We suggest an approach based on the following main procedures: usage of comprehensive internal security repository and open security databases; generation of attack trees considering service dependency graphs and zero-day vulnerabilities; application of anytime algorithms to provide near real-time attack modeling; usage of attack graphs to predict possible malefactor's actions; calculation of a multitude of security metrics, attack and response impacts; interactive decision support to select the security solutions. The main difference of the offered approach from the already suggested ones is the integration of these functionalities in one component to achieve better results in near real time effective attack modeling and security evaluation. The approach novelty consists also in the way of modeling attacks (we use a multi-level model of attack scenarios based on known and unknown (zero day) vulnerabilities) and applying constructed attack graphs and service dependencies to determine a family of security metrics and comprehensive evaluation of security properties.

In the paper we discuss the architectural solution of the proposed Attack Modeling and Security Evaluation Component (AMSEC) as one of the important SIEM subsystem and the techniques used to realize main AMSEC functionality. To illustrate these architecture and techniques we developed a software prototype and carried out experiments for different case-studies. The prototype architecture and the results of the experiments are presented.

The rest of the paper is organized as follows. In Section II, we review the related work. Section III discusses the AMSEC framework. In Section IV, we describe the AMSEC implementation. Section V presents experiments. In conclusion we analyze the paper results and provide insight into our future research.

II. RELATED WORK

There are a lot of papers, which consider different approaches to attack modeling and security evaluation taking into account various classes of attacks. We analyze briefly current state-of-the-art in representation of attack scenarios and malefactors, generation of attack graphs, determining security metrics, combining service dependency graphs with attack graphs, and representing zero day attacks.

In [8, 11, 24] attacks are described and modeled in a structured and reusable tree-based form. In [11] a high-level conceptual model of attack based on the intruder's intent (attack strategy) is presented. The paper determines intrusion intention as the goal-tree. The ultimate goal of intrusion corresponds to the root node. Lower level nodes represent alternatives or ordered sub-goals in achieving the upper node/goal. The logical constructs are used for representation of temporal sequences of intrusion intentions. The comprehensive work using the so-called tree-based approach is proposed in [24]. This paper describes means for documenting attacks in a form of attack trees.

One of the most important problems in security analysis is the malefactors' classification and model construction. In [9] the task of modeling and simulation of intelligent, reactive attackers is described. The suggested computer

978-0-7695-4865-4/12 $26.00 © 2012 IEEE
DOI 10.1109/GreenCom.2012.24

network attack model uses an action representation based on the GOLOG situation calculus [18] and goal-directed procedure invocation. Goldman has designed components of a stochastic attack simulator which can simulate some goal-directed attacks on a network.

Different approaches, which use attack graphs and trees for security analysis, have been suggested. S. Hariri et al. [10] calculate global metrics to analyze and proactively manage the effects of complex network faults and attacks. S. Noel, S. Jajodia et al. [25, 31] propose a risk based technique based on determining the minimum-cost network hardening via exploit dependency graphs.

I. Kotenko and M. Stepashkin [15-17] are focused on security metrics computations based on attack graph representation of malefactor behavior.

R. Lippmann and K. Ingols [29] propose to use attack graphs to detect firewall configuration defects and host critical vulnerabilities. Later this approach was extended by taking into modern network attacks threats (zero-day exploits and client-side attacks) and countermeasures (intrusion prevention systems, personal firewalls, and host-based vulnerability scanners) [12].

J. Ryan and D. Ryan [27] suggest an approach to calculate metrics based on failure-time analysis. L. Wang, S. Jajodia et al. [32, 33] propose an approach to calculate attack resistance metrics based on probabilistic scores by combining CVSS scores [6]. N. Kheir et al. [14] suggest an implementation of confidentiality, integrity and availability metrics using the notion of privilege, which is inspired by access permissions within access control policies.

There is a new trend of research in attack modeling, which is to combine attack graph models and service dependency models. In their essence, attack graphs represent possible attacker actions in the light of current system configuration. Meanwhile, they do not represent service dependencies and their underlying connection requirements. N. Kheir et al. [13] propose to extend the use of CVSS metrics in the context of intrusion response, by supplying this metric with dynamic information about system configuration and service dependencies structured within dependency graphs. The dependency graph is further used to evaluate the overall impact of an attack, thus replacing the informal environmental parameters in the CVSS vector. Nonetheless, the problem with this approach is that it does not provide clear evidence on how to interface service dependency graphs with attack graph models.

The analysis of network security against unknown zero day attacks is also a relatively new topic of research. Zero day attacks can be defined as attacks which use unknown vulnerabilities.

E. Bursztein [1] extends the security analysis approach, based on game theory, by taking into account zero day exploits. L. Williams [35] presents a practical realization of the approach to calculate the possible number of zero day vulnerabilities. M. McQueen et al. [22] attempt to evaluate the total number of possible zero day vulnerabilities for one day. K. Ingols et al. [12] suggest ordering different applications by the seriousness of consequences of having a single zero day vulnerability. L. Wang et al. [33] propose a security metric called k-zero day safety. It is based on how many unknown vulnerabilities are required to compromise a network asset, regardless of the type of vulnerabilities.

III. Main Framework

According to the analysis of state-of-the-art in attack modeling we selected the following key elements to be included in the architectural solution of AMSEC as part of the SIEM-system (Fig. 1):

- Comprehensive security data repository;
- Effective attack tree and service dependencies generation techniques based on TVA (Topological Vulnerability Analysis) approach which enumerates potential sequences of exploits of known vulnerabilities to build attack graphs;
- Attack graph generation considering both known and zero-day vulnerabilities;
- Usage of anytime algorithms for near-real time attack sub-graph (re)generation and analytical modeling;
- Stochastic analytical modeling;
- Combined usage of attack graphs and service dependency graphs;
- Security metric calculation, including attack impact, response efficiency, response collateral damages, attack potentiality, attacker skill level assessment, etc.
- Interactive decision support to select the solutions on security measures/tools by defining their preferences regarding different types of requirements (risks, costs, benefits) and setting trade-offs between high-level security objectives.

To bind the key elements we developed the following generalized architecture of AMSEC (Fig. 2). The lines in the Fig. 2 reflect the links between AMSEC's modules and other SIEM components (Correlation Engine, Security Event Modeler, Predictive Security Analyzer, Decision Support & Reaction System).

Fig. 3 illustrates the main data flows in AMSEC and its input and output data. We suppose that AMSEC can function in two modes: configuration (or design) and exploitation.

In the first mode the AMSEC operates in non real-time with the model of analyzed computer network (system) based on design specifications of computer network configuration and security policy, producing the list of weak network places, possible zero-day vulnerabilities, generating the set of attack trees.

The exploitation mode is a real-time or near real-time one, in this mode AMSEC adjusts existing attack trees and malefactor model, predicts malefactor's actions and generate countermeasures.

The brief descriptions of the AMSEC's modules and their functions are given below.

Network interface supports interaction with external environment (sending requests to external databases and communicating with data sources).

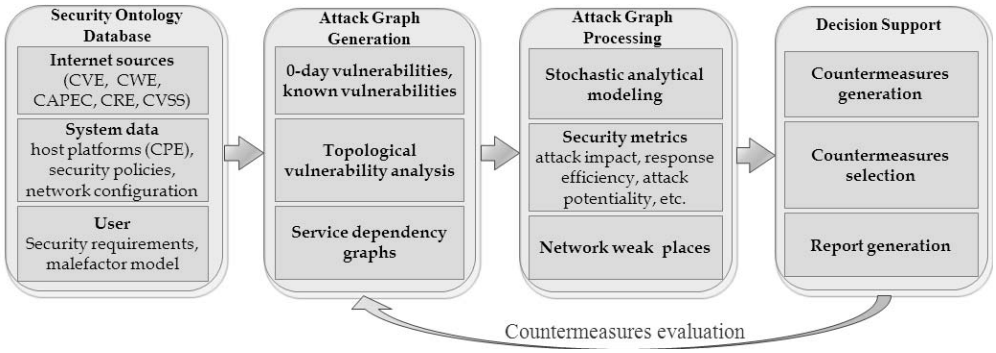

Figure 1. Attack Modeling and Security Evaluation Framework

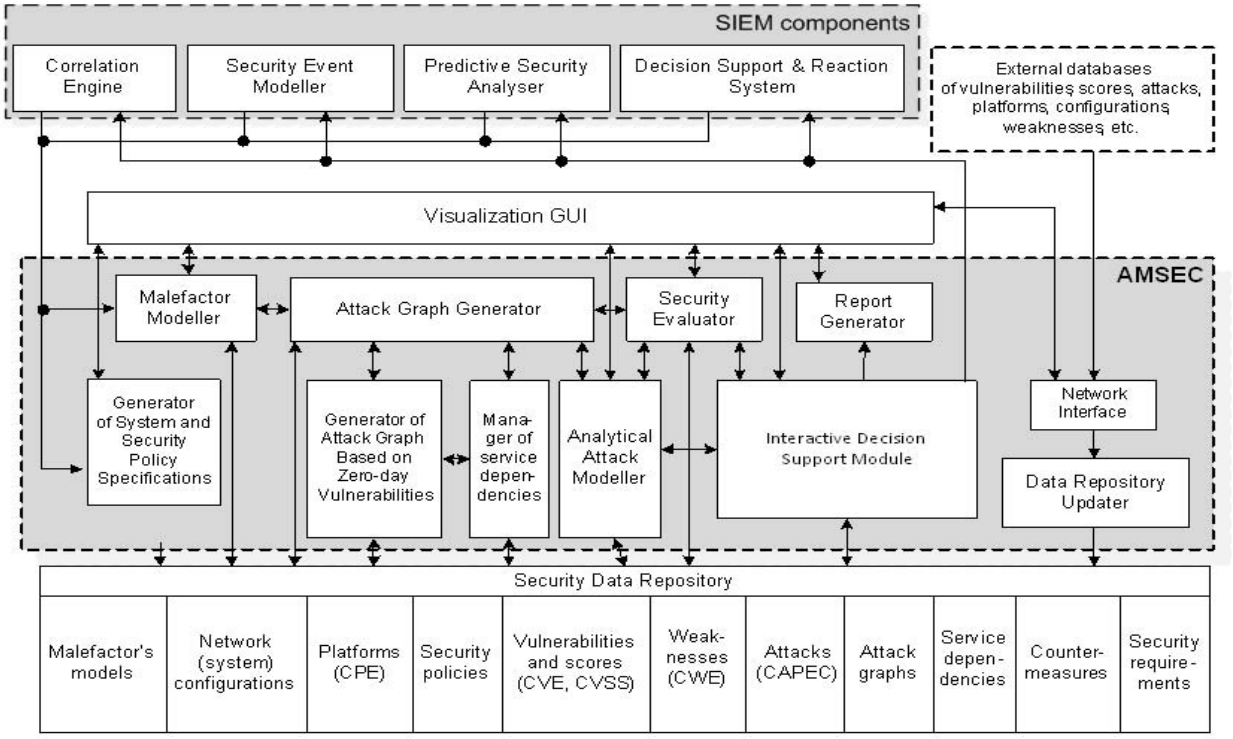

Figure 2. Generalized architecture of AMSEC.

Interactive decision support module provides the user (decision maker) with the ability to select the solutions on countermeasures by defining their preferences regarding different types of requirements and setting trade-offs between objects. Decision support can include three phases: setting feasible security solutions (security measures/tools); identification of efficient (Pareto-optimal) security solutions; selection (generation) of the final solution.

Generator of system and security policy specification converts the information about network configuration and security policy received from the data collection and correlation components or user into internal representation. It is supposed, that at the design stage, this information is specified on special System Description Language and Security Policy Language. Used specifications of the analyzed network (system) and the security policy should describe network components with the necessary degree of detail; for example, the used software should be set in the form of names and versions.

Data repository updater downloads the open databases of vulnerabilities, attacks, configuration, weaknesses, platforms, countermeasures, etc., for example, National Vulnerability Database (NVD) [26], Common Vulnerabilities and Exposures (CVE) [4], Common Attack Pattern Enumeration and Classification (CAPEC) [2], Common Platform Enumeration (CPE) [3], and then translates them into the AMSEC security data repository.

Reports generator shows vulnerabilities detected by AMSEC, represents "weak" places, generates recommendations on strengthening the security level, etc.

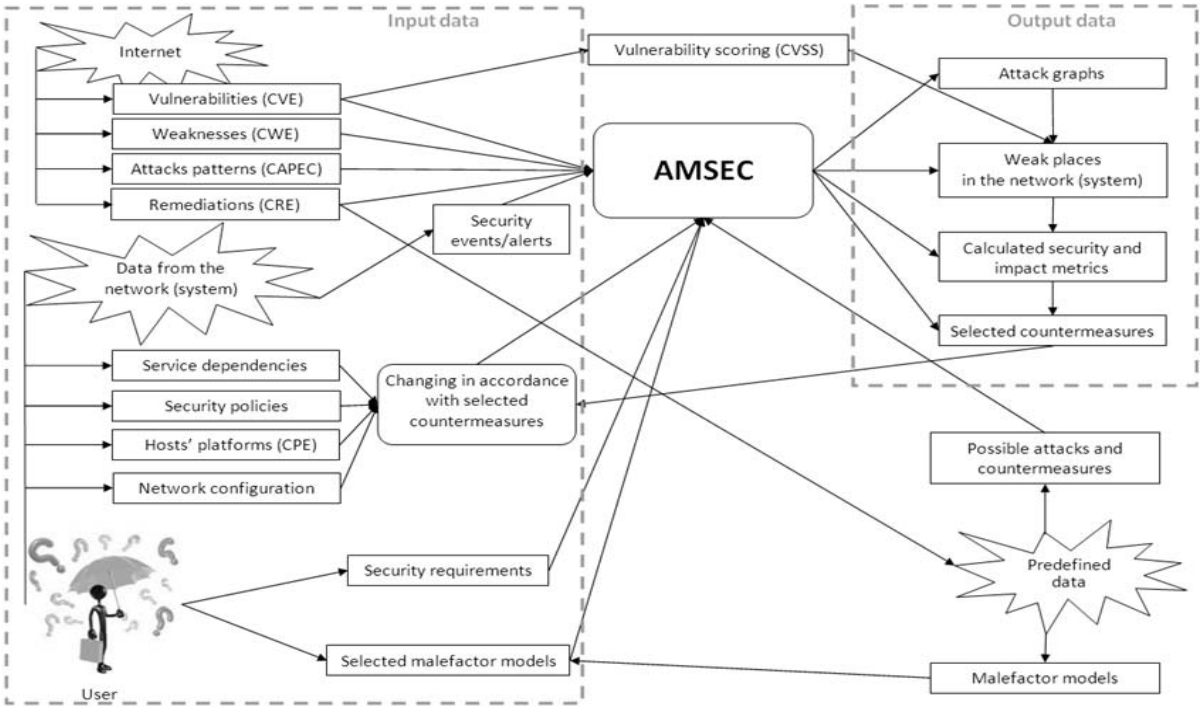

Figure 3. Main data flows in AMSEC.

Security repository is a hybrid (relational, XML-based and triplet-based) data storage which contains information necessary for attack graph generation and analysis. We suggest to use a set of MSM related standards [20] or other related standards for the common enumeration, expression and reporting of cyber-security-related information as the basis for the design of the common security repository.

The input data consist of two types of data: data obtained from external sources in the Internet (vulnerabilities, weaknesses, and attack patterns [2, 4, 6, 7]), and data obtained by analyzing the network (system) and generated by scanning tools and users.

The network events and alerts generated by data collection and correlation components, malefactor's model, network configuration, hosts' platform, service dependencies, possible countermeasures, security requirements, policies and configuration denote to the second type of the input data.

We understand output data as data obtained from the attack modeling component in the result of simulation: attack graphs, security and impact metrics, selected countermeasures, and elements of the tested network, breaking of which leads to the greatest damage (weak places).

Malefactor Modeler is responsible for malefactor modeling and is used on both design and exploitation stage of the AMSEC operation. On the first phase it is used to build the set of all possible attack graphs using preset characteristics of malefactor (the malefactor profile) which are determined by the user. Later on the second phase it allows predicting the possible characteristics of the malefactor according to the actions fulfilled.

The malefactor's actions are mapped to the previously generated set of attack graphs and thus it is possible to predict his/her next actions in real time mode. Besides the information about attacker's actions is used to re-evaluate dynamically his/her characteristics (skills, initial set of access permissions, etc.) that helps to define more precisely the attacker's strategy. Thus, assessing malefactor's characteristics enables to adapt the severity of an attack with the attacker profile. Malefactor's skill level could assist, among other metrics, the response decision support (such as allowing a more severe response in case of a highly skilled attacker).

Attack Graph Generator is responsible for attack graph building. We use TVA to generate attack graph, this technique is based on enumeration of potential sequences of attack actions (using exploits of known and zero-day vulnerabilities). Also we are implementing two types the analysis – backward and forward depending on state of search start (final or initial) [29, 30].

Attack Graph Generator operates in conjunction with *Manager of Service Dependencies and Generator of Attack Graph Based on Zero-day Vulnerabilities* to obtain more precise results in attack modeling.

To get more precise information about intrusion impact and response impact propagation we use the service dependencies graph in our approach. This solution allows assessing the impact propagation for the intrusion and response on the basis of CVSS [6] and giving quantitative metrics [13, 14]. We expand common model for attack/defense analysis by adding new object "Service" with specific properties that describe trust relationships between network objects. Thus we define additional service layer.

The algorithm of attack graph generation was changed according to the modifications in the attack model. We add an additional component – *Manager of Service Dependencies* which operates with service dependencies. For every atomic attack (each step in the attack graph) this component creates an additional service dependency graph.

The usage of service dependency graphs makes it possible to exclude information about attack impacts from the attack graph and to use the dependency graph in order to simulate impacts and obtain a dynamic evaluation of an attack impact.

In our approach we also take to into account zero-day vulnerabilities to generate attack graph. To do this we modify the approach suggested in [12] by adding additional characteristics which define probability of existence of the zero day vulnerability. The main idea is to automate process of selection of hosts which are likely to have zero day vulnerabilities then others (instead of manual search). *Generator of Attack Graph Based on Zero-day Vulnerabilities* includes two sub-components – Zero-day Existence Analyzer and *Graph Generator*.

The set of vulnerable hosts/applications serves as an input for the *Graph Generator* that outputs attack graph. In addition to the attack graph generator based on the approach of [12] we consider a similar approach suggested in [28]. In this model-based approach the resilience of an information infrastructure against attacks to unknown (zero-day) vulnerabilities is analyzed by definition of a generic new vulnerability for each installed product.

Security Evaluator is responsible for qualitative and quantitative assessment of the system security. For qualitative express assessment of the network security we planning to use several approaches which are based on different security metrics, risk analysis and security evaluation techniques.

The approach of qualitative express assessment of network security level uses the following metrics as basic ones: *Criticality*(*h*) - criticality level of host *h*; *Severity*(*a*) - criticality level of attack action *a*; *Mortality*(*a,h*) - damage level caused by attack action, taking into account the criticality level of host; *Mortality*(*S)* and *Mortality*(*T*) - damage level of route *S* and threat *T*; *AccessComplexity*(*a*), *AccessComplexity*(*S*), *AccessComplexity*(*T*) - "access complexity" of attack action *a*, route *S* and threat *T*; *Realization*(*T*) - admissibility of threat realization; *RiskLevel*(*T*) - risk level of threat *T*; *SecurityLevel* – general security level of the computer network.

IV. Implementation

By now a prototype of AMSEC, which can generate possible attack trees for a predefined network and evaluate the network security level, was implemented. It contains three basic components: *VDBUpdater*, *Network Constructor* and *Security Level Evaluator*. Additionally the prototype includes the MySQL database as a common repository.

VDBUpdater allows updating the internal database of known vulnerabilities, using information obtained from National Vulnerability Database [26]. It consists of two components: the component intended to preload the XML representation of the NVD database and the repository updater which loads XML representation of the NVD database to the local or remote database.

Network Constructor aims to create and modify network models. It includes Generator of system and security policy specification (allows users creating models of tested computer networks) and Data controller (checks selected software and hardware to match NVD dictionary). Using *Network Constructor*, users can perform the following tasks: viewing in graphical form (as a graph) the network structure; setting and modifying the network metadata (name, date and time of creation, etc.); creating, modifying and deleting network elements (workstations, switches, etc.); setting and modifying the metadata of network elements (name, location, level of criticality, etc.); creating and deleting links between network elements.

Security Level Evaluator generates attack graphs, makes topological vulnerability analysis, enumerates potential sequences of exploits of known vulnerabilities and evaluates the security level of the network. It consists of the following components: the common attack graph generator, the security evaluator, which determines security status and the report generator, which generates reports consisting of a list of operations performed by the attacker as well as a list of detected vulnerabilities and security metrics.

Security Level Evaluator allows users viewing the specification of the tested computer network, the attacker knowledge about the tested network, the attack graph, the event log, including all actions preformed by attacker, all detected vulnerabilities and the results of calculating the particular security metrics and the common security level of the tested network. Inputs of this component are the file containing the network data in predefined format, database of current vulnerabilities, and host(s) where malefactor is situated. Output is an attacks tree and security metrics calculated.

V. Experiments

A set of experiments with the prototype of AMSEC was conducted. The prototype makes use of scenario "Critical Infrastructure Process Control (Dam)" [21]. In this paper we present the results of security analysis of the dam infrastructure.

The features of the dam infrastructures are strictly related to the aims they are conceived for; mostly dams are used for water supplying, hydroelectric power generation, irrigation, water activities and wildlife habitat granting. In the case study "Critical Infrastructure Process Control (Dam)" the reference system architecture involves typical SCADA components. We can identify three main groups of components in this system: control devices, I/O devices and a SCADA gateway. Thus, we outline the following network elements necessary for attack modeling: sensors; network hardware (firewall, router, etc.); computers with installed software (web-server, application servers, database servers, users' computers); links between the network elements (wired, wireless).

Obviously the security of the dam depends primarily on integrity and authenticity of the data received from sensors. That's why the most misuse cases are associated with compromise of the sensors. Also malefactor can try to block the dam control commands, fulfill hazardous water release operations and misuse visualization stations. To accommodate all possible attacks we outlined the following types of malefactors by their physical location: on the dam territory – Malefactor 1; on the territory of the control station – Malefactor 2; outside of the controlled network (access via the Internet) – Malefactor 3.

Fig. 4 illustrates the topology of the tested network and possible attacker's location.

Let us consider the experiments where we chose the Malefactor 3, i.e. a malefactor located outside the controlled network. Thus, the initial position of the malefactor is one of the computers in the Visualization Users group, where he/she has unlimited rights. Since the malefactor is an external user for the controlled network, then he/she has no rights in the network. To make clearer the illustration of the AMSEC prototype possibilities we consider in the paper a case with the following software for network hosts: OS Windows Server 2003 is installed on all hosts, DBMS MySQL 5.0 is installed on the host Database2, Apache HTTP Server 1.3.6 is installed on the host Visualization Web Server.

After constructing the attack graph, the AMSEC provides the following information: the malefactor knowledge after all possible attacks, the attack tree in the graphic form and the log of the malefactor's actions.

Fig. 5 illustrates different attacks traces that attacker can perform in the tested network.

The attacker, carrying out attack actions, is located in the centre of the spherical representation. The other icons are as follows: "A" – an attack action, "S" – scenario which does not use vulnerabilities (for example, host discovery (PING)), "V" – an attack action which exploits some vulnerability.

According to the attack graph the chain of malefactor's actions and their results are as follows: (1) Detection of nodes connected with the initial malefactor host. Visualization Web Server host is detected. (2) Detection of the software installed on the Visualization Web Server host. Windows Server 2003 is detected. (3) Usage of the vulnerability CVE-2007-0214 [5]. Malefactor compromises of the Control Visualization Web Server. (4) Detection of the nodes connected with the Visualization Web Server. Application Server host is detected, etc.

According to the suggested metrics the security level of the tested network is evaluated. For each node the criticality level is determined, for example for the nodes "Visualization Users" and "Application Server" it is LOW while for the node "Firewall" it is HIGH. For each attacker's action and each possible attack route the security metrics *Access Complexity* (AC) and *Mortality* (M) are calculated.

Here we present the values of these metrics computed for the route "Visualization Users - Firewall". To gain access to the Firewall the attacker needs to fulfill the following actions: ping Visualization Web Server (M:LOW; AC:LOW) → detect OS Visualization Web Server (M:LOW; AC:LOW) → use CVE-2007-0214 Visualization Web Server (M:MEDIUM; AC:MEDIUM) → ping Application Server2 (M:LOW; AC:LOW) → detect OS Application Server2 (M:LOW; AC:LOW) → use CVE-2007-0214 Application Server2 (M:MEDIUM; AC:MEDIUM) → ping Firewall (M:LOW; AC:LOW) → determine OS Firewall (M:LOW; AC:LOW).

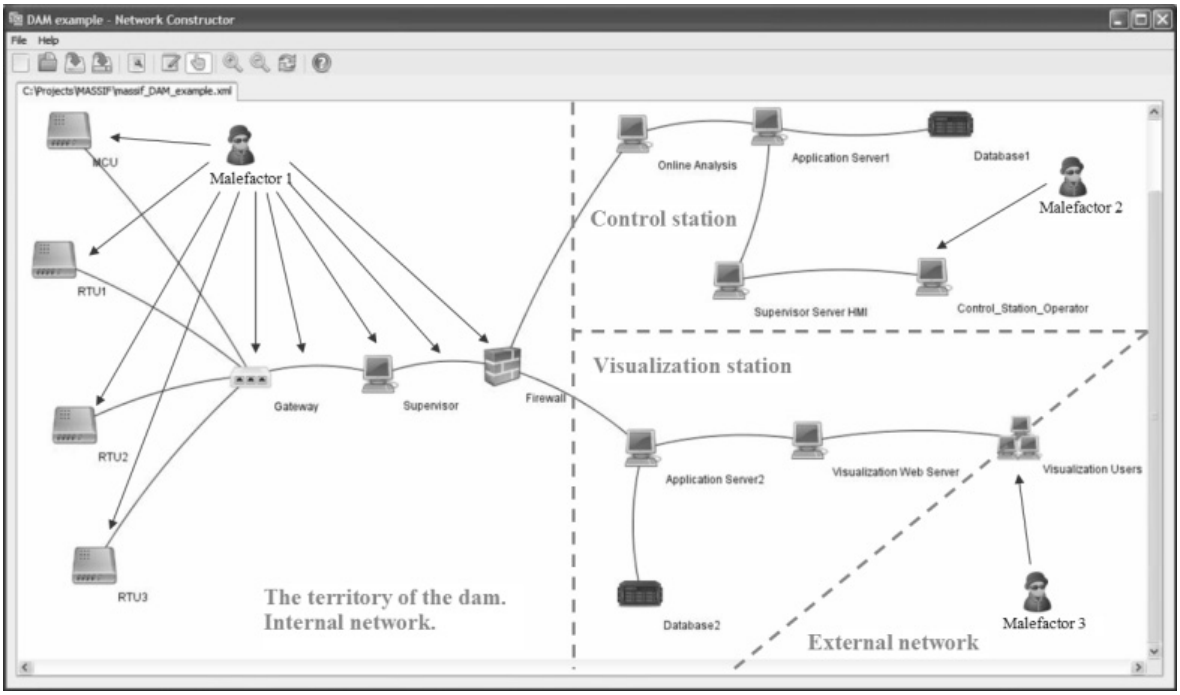

Figure 4. Dam network topology and attacker's locations.

Thus, the Route parameters *Access Complexity* and *Mortality* equal LOW. These metrics form the basis for the general network level evaluation. In this use case the Security Level is ORANGE, what means that countermeasures need to be implemented.

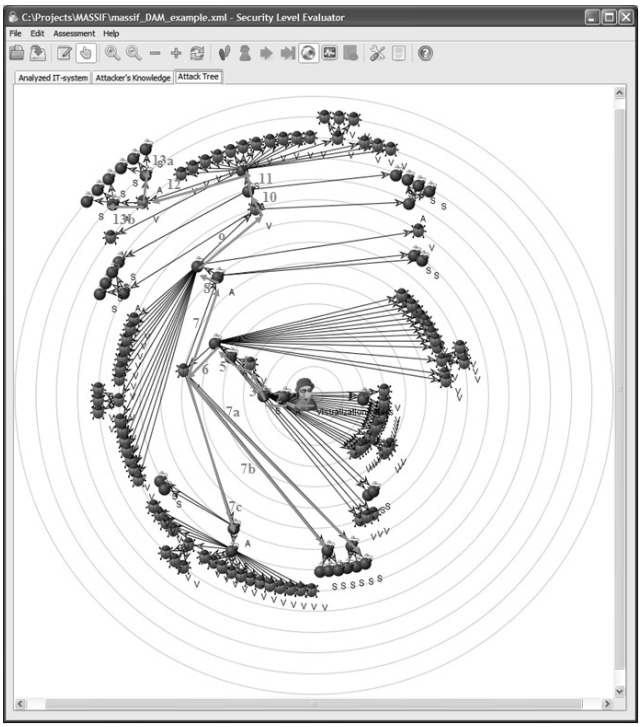

Figure 5. Example of an attack graph.

Fig. 6 depicts a fragment of the log of malefactors' actions. Fig. 7 shows the report on security evaluation.

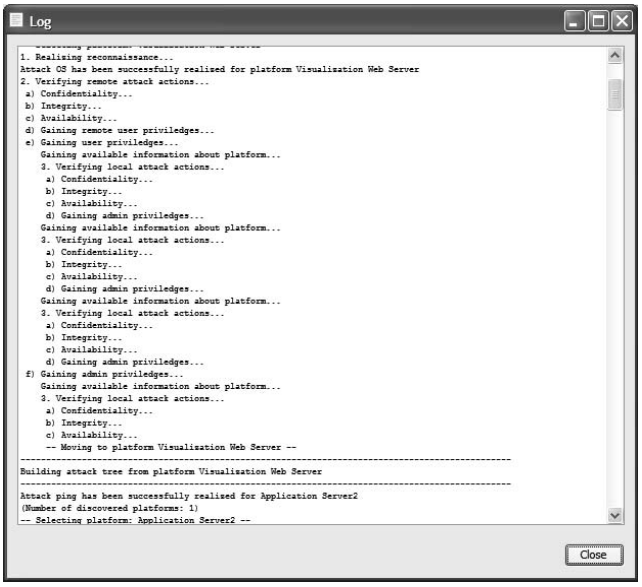

Figure 6. Log of the attack graph building.

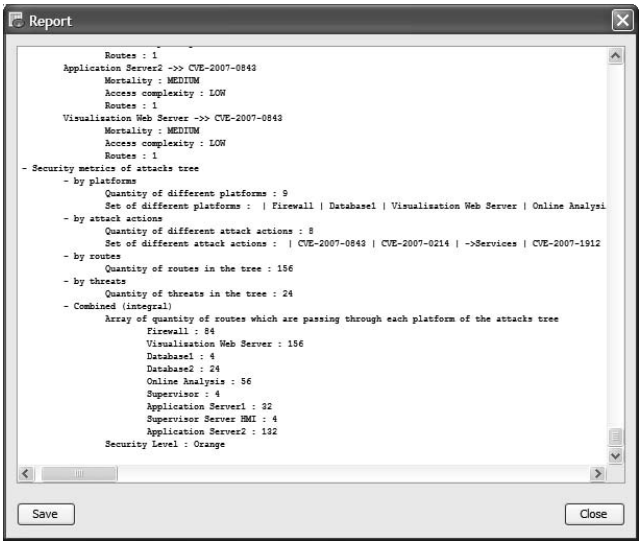

Figure 7. Report on security evaluation.

VI. CONCLUSION

In the paper we presented our approach to the attack modeling and security evaluation. It has following peculiarities:

- Usage of integrated family of different models based on expert knowledge, including malefactor's models, multilevel models of attack scenarios, building attack graph, specifying service dependencies, security metrics evaluation;
- Taking into account diversity of malefactor's positions, intentions and experience levels;
- Usage (during construction of common attack graph) the parameters of computer network configuration, the rules of security policy, fixed events and alerts; possibility of estimating the influence of configuration and policy data on the security level;
- Taking into account not only attack actions which use known vulnerabilities, but zero days attacks and traditional actions of legitimate users and reconnaissance actions;
- Possibility of investigating various threats for different network resources;
- Possibility of detection of bottlenecks – weak places (hosts and applications responsible for the most serious attack actions, routes and threats);
- Interactive decision support to select the solutions on security measures/tools by defining their preferences regarding different types of requirements (risks, costs, benefits) and setting trade-offs between high-level security objectives; possibility of querying the system in the "what-if" way, for example, how the general security level will change if the certain parameter of security policy is changed;
- Usage of updated databases, for example, NVD, CVE, CAPEC, CPE, CCE. Usage of CVSS and qualitative techniques of risk analysis.

We also described an implemented prototype of the AMSEC, which can generate a possible attack tree for a predefined network and a simple experiment was considered.

The suggested approach allows us to achieve more accurate evaluation of network security in contrast to other particular approaches. The usage of the near real time algorithms also enables to get the results faster than existing approaches if it is needed, on the other hand fast results could have less precision.

The future steps of the research will be devoted to detailed elaboration of all AMSEC components. One of the important research issues is development of techniques which can cope with large networks, such as those in enterprise infrastructure. Also it is planned to optimize the generation of attack trees through the use of the ontology based repository, to expand the list of parameters, characterizing the hosts and the network, to improve the malefactor model, and to add currently unrealized components.

ACKNOWLEDGMENT

This research is being supported by grant of the Russian Foundation of Basic Research (project #10-01-00826-a), Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (contract #2.2), State contract #11.519.11.4008 and partly funded by the EU as part of the SecFutur and MASSIF projects.

REFERENCES

[1] E.Bursztein, “Extending Anticipation Games with Location, Penalty and Timeline”, LSV, ENS Cachan, CNRS, INRIA, France, 2008.

[2] CAPEC. Common Attack Pattern Enumeration and Classification. http://capec.mitre.org/, 2012.

[3] CPE. Common Platform Enumeration. http://cpe.mitre.org/, 2012.

[4] CVE. Common Vulnerabilities and Exposures. http://cve.mitre.org/, 2012.

[5] CVE-2007-0214, 2007. Vulnerability Summary for CVE-2007-0214. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-0214, 2012.

[6] CVSS. Common Vulnerability Scoring System. http://www.first.org/cvss/, 2012.

[7] CWE. Common Weakness Enumeration. http://cwe.mitre.org/, 2012.

[8] J.Dawkins, C.Campbell, J.Hale, “Modeling network attacks: Extending the attack tree paradigm”, Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Johns Hopkins University, 2002.

[9] R.P.Goldman, “A Stochastic Model for Intrusions”, Lecture Notes in Computer Science”, V.2516. Springer Verlag, 2002, pp. 199-218.

[10] S.Hariri, G.Qu, , T.Dharmagadda, M.Ramkishore, C.S.Raghavendra, “Impact Analysis of Faults and Attacks in Large-Scale Networks”, IEEE Security and Privacy, vol.1, 2003, pp. 49-54.

[11] M.-Y.Huang, T.M.Wicks, “A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis”, Computer Networks, vol. 31, New York, NY, USA, 1999, pp. 2465-2475.

[12] K.Ingols, M.Chu, R.Lippmann, S.Webster, S.Boyer, “Modeling modern network attacks and countermeasures using attack graphs”, 2009 Annual Computer Security Applications Conference (ACSAC ’09), Washington, D.C., USA, IEEE Computer Society, 2009.

[13] N.Kheir, H.Debar, N.Cuppens-Boulahia, F.Cuppens, J.Viinikka, “Cost evaluation for intrusion response using dependency graphs”, IFIP International Conference on Network and Service Security (N2S), IEEE, Paris, France, 2009, pp. 1-6.

[14] N.Kheir, N.Cuppens-Boulahia, F.Cuppens, H.Debar, “A service dependency model for cost-sensitive intrusion response”, ESORICS 2010, Athens, Greece, 2010, pp. 626-642.

[15] I.Kotenko, M.Stepashkin, “Network Security Evaluation based on Simulation of Malefactor's Behavior”, International Conference on Security and Cryptography (SECRYPT-2006), Portugal, 2006.

[16] I.Kotenko, M.Stepashkin, “Attack Graph based Evaluation of Network Security”, Lecture Notes in Computer Science, vol.4237, 2006, pp. 216-227.

[17] I.Kotenko, M.Stepashkin, E.Doynikova, “Security Analysis of Computer-aided Systems taking into account Social Engineering Attacks”, PDP 2011, Los Alamitos, California. IEEE Computer Society, 2011.

[18] H.J.Levesque, R.Reiter, Y.Lesperance, F.Lin, R.B.Scherl, “GOLOG: A Logic Programming Language for Dynamic Domains”, Journal of Logic Programming, vol. 31, No.1-3, 1997.

[19] R.Lippmann, K.Ingols, “Validating and Restoring Defense in Depth Using Attack Graphs”, MILCOM 2006. Washington, DC, 2006.

[20] MSM. Making Security Measurable. http://measurablesecurity.mitre.org/ index.html, 2012.

[21] MASSIF. Massif project. http://www.massif-project.eu, 2012.

[22] M.McQueen, T.McQueen, W.Boyer, M.Chaffin, “Empirical estimates and observations of 0-day vulnerabilities”, Hawaii International Conference on System Sciences, 2009.

[23] D.R.Miller, Sh.Harris, A.A.Harper, S.VanDyke, Ch.Black, “Security Information and Event Management (SIEM) Implementation”, McGraw–Hill Companies, 2011.

[24] A.P.Moore, R.J.Ellison, R.C.Linger, “Attack Modeling for Information Security and Survivability”, Technical Note CMU/SEI-2001-TN-001. Survivable Systems, 2001.

[25] S.Noel, S.Jajodia, B.O’Berry, M.Jacobs, “Efficient minimum-cost network hardening via exploit dependency graphs”, ACSAC’03, 2003, P. 86.

[26] NVD. National Vulnerability Database. http://nvd.nist.gov/, 2012.

[27] J.Ryan, D.Ryan, “Performance metrics for information security risk management”, IEEE Security and Privacy, vol 6, 2008, pp. 38-44.

[28] R.Rieke, “Abstraction-based analysis of known and unknown vulnerabilities of critical information infrastructures”, International Journal of System of Systems Engineering (IJSSE). InderScience. vol. 1, 2008, pp. 59-77.

[29] O.Sheyner, J.Haines, S.Jha, “Automated generation and analysis of attack graphs”, 2002 IEEE Symposium on Security and Privacy. Berkeley, California, 2002.

[30] L.P.Swiler, C.Phillips, D.Ellis, S.Chakerian, “Computer-attack graph generation tool”, Proc. DARPA Information Survivability Conference, Proceedings. Anaheim, CA, vol. 2, pp. 307-321, 2001.

[31] L.Wang, S.Noel, S.Jajodia, “Minimum-cost network hardening using attack graphs”, Computer Communications, vol. 29, 2006.

[32] L.Wang, T.Islam, T.Long, A. Singhal, S. Jajodia, “An attack graph-based probabilistic security metric”, Proc. of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security. Springer-Verlag Berlin, pp. 283-296.

[33] L.Wang, S.Jajodia, A.Singhal, S.Noel, “k-Zero Day Safety: Measuring the Security Risk of Networks against Unknown Attacks”, ESORICS'10. Springer-Verlag, Berlin, Heidelberg, 2010.

[34] L.Wang, J.N.Whitley, R.C.W.Phan, D.J.Parish, “Unified Parametrizable Attack Tree”, International Journal for Information Security Research, vol.1(1), 2011.

[35] L.Williams, “GARNET: A Graphical Attack Graph and Reachability Network Evaluation Tool”, Proc. of the 5th international workshop on Visualization for Computer Security, Springer-Verlag Berlin, 2008.