International Transactions on Systems Science and Applications
Volume 8 ▪ December 2012 ▪ pp. 129-147
sai: itssa.0008.2012.041

ISSN 1751-1461 (print)
ISSN 2051-5642 (online)

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
Email: {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 applicable for future systems of the Internet of Things. 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, the results of experiments carried out, and comparison of suggested attack modeling and security evaluation solutions with existing ones.

Keywords: attack modeling, security evaluation, SIEM, attack graph, service dependences, zero day vulnerabilities.

1. Introduction

There are many different reasons for security violations occurring in computer networks: security policy errors, vulnerabilities, incorrect configurations, etc. Malefactors can use different vulnerabilities and bottlenecks of network configuration and security policy and perform different penetration strategies. These strategies are directed to different network resources and include various assault actions chains. Malefactors can step-by-step compromise network hosts and realize different security threats.

Therefore, in SIEM systems the security administrator should check whether network configuration parameters and security procedures provide the necessary security level. Moreover, at exploitation stage, current security events and alerts should be taken into account, the configuration of computer networks can be changed, new vulnerabilities can be discovered, new attack exploits can be developed, new services can be added, and it is necessary continually to perform network monitoring, analyze available vulnerabilities and evaluate security level.

The complexity of computer network security management causes the necessity to develop powerful automated security analysis components which can be important subsystems of Security Information and Event Management (SIEM) systems [16, 36]. 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.

The paper discusses 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 comparison of suggested attack modeling and security evaluation solutions with existing ones is fulfilled.

The rest of the paper is organized as follows. In Section 2, the related work is reviewed. Section 3 discusses the AMSEC framework. In Section 4, the AMSEC implementation is described. Section 5 presents examples of experiments. Section 6 is a comparison with related systems. In conclusion the paper results are analyzed and insight into the future research is provided.

2. 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 [1] [15] [26] attacks are described and modeled in a structured and reusable tree-based form. In [26] 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 [1]. 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 [36] the task of modeling and simulation of intelligent, reactive attackers is described. The suggested computer network attack model uses an action representation based on the GOLOG situation calculus [11] 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. [38] calculate global metrics to analyze and proactively manage the effects of complex network faults and attacks. S. Noel, S. Jajodia et al. [21] [39] propose a technique based on determining the minimum-cost network hardening via exploit dependency graphs.

I. Kotenko and M. Stepashkin [12] – [14] are focused on security metrics computations based on attack graph representation of malefactor behavior.

R. Lippmann and K. Ingols [33] 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) [17].

J. Ryan and D. Ryan [16] suggest calculating metrics based on failure-time analysis. L. Wang, S. Jajodia et al. [22] [20] propose to calculate attack resistance metrics based on probabilistic scores by combining CVSS scores [7]. N. Kheir et al. [30] 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. [29] 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 [10] extends the security analysis approach, based on game theory, by taking into account zero day exploits. L. Williams [23] presents a practical realization of the approach to calculate the possible number of zero day vulnerabilities. M. McQueen et al. [25] attempt to evaluate the total number of possible zero day vulnerabilities for one day. K. Ingols et al. [17] suggest ordering different applications by the seriousness of consequences of having a single zero day vulnerability. L. Wang et al. [20] 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.

3. 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 the AMSEC as part of the SIEM system (Figure 1):

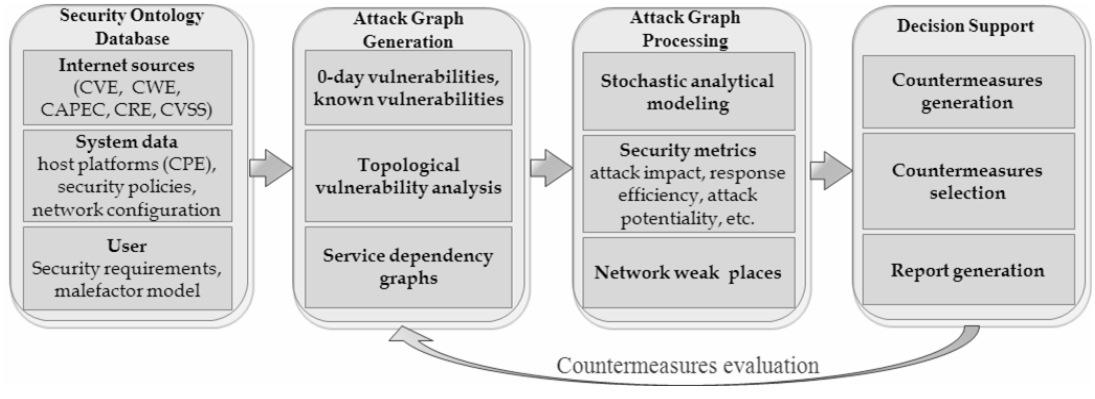

Figure 1. Attack Modeling and Security Evaluation Framework.

(i) Comprehensive security data repository;
(ii) Effective attack tree and service dependencies generation techniques based on the TVA (Topological Vulnerability Analysis) approach which enumerates potential sequences of exploits of known vulnerabilities to build attack graphs;

(iii) Attack graph generation considering both known and zero-day vulnerabilities;
(iv) Usage of anytime algorithms for near-real time attack sub-graph (re)generation and analytical modeling;
(v) Stochastic analytical modeling;
(vi) Combined usage of attack graphs and service dependency graphs;
(vii) Security metric calculation, including attack impact, response efficiency, response collateral damages, attack potentiality, attacker skill level assessment, common security level, etc.;
(viii) 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 (Figure 2).

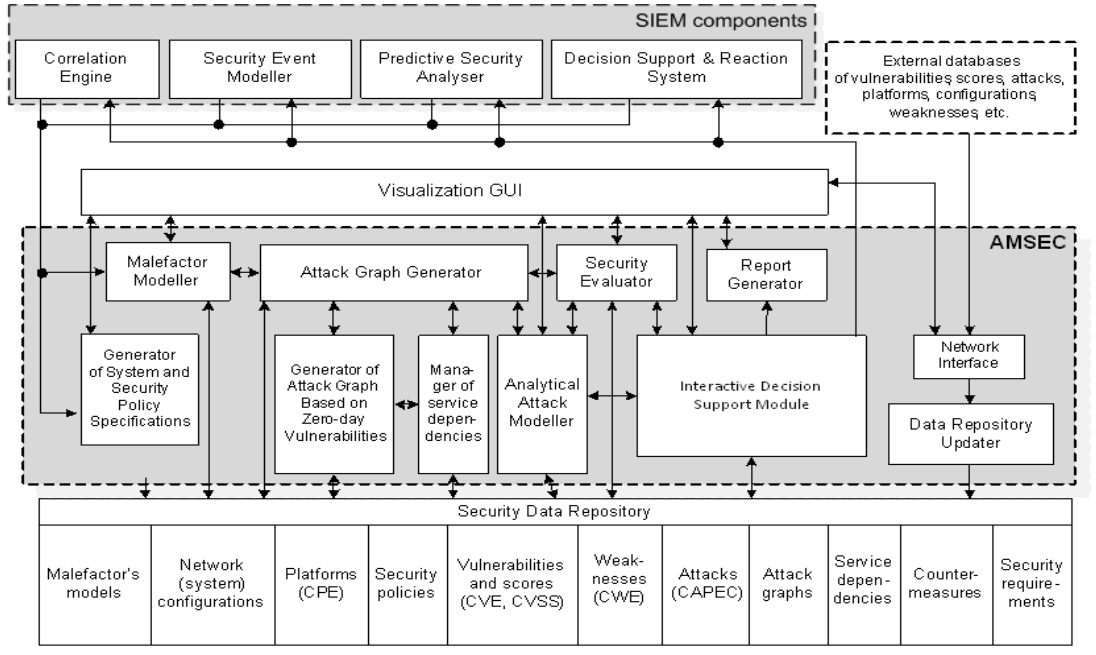

Figure 2. Generalized architecture of the AMSEC.

The lines in Figure 2 reflect the links between the AMSEC's modules and other SIEM components (Correlation Engine, Security Event Modeler, Predictive Security Analyzer, Decision Support and Reaction System).

Figure 3 illustrates the main data flows in AMSEC and its input and output data. We suppose that the AMSEC can function in two modes: (1) configuration (or design) and (2) 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 the AMSEC adjusts existing attack trees and malefactor model, predicts malefactor's actions and generate countermeasures.

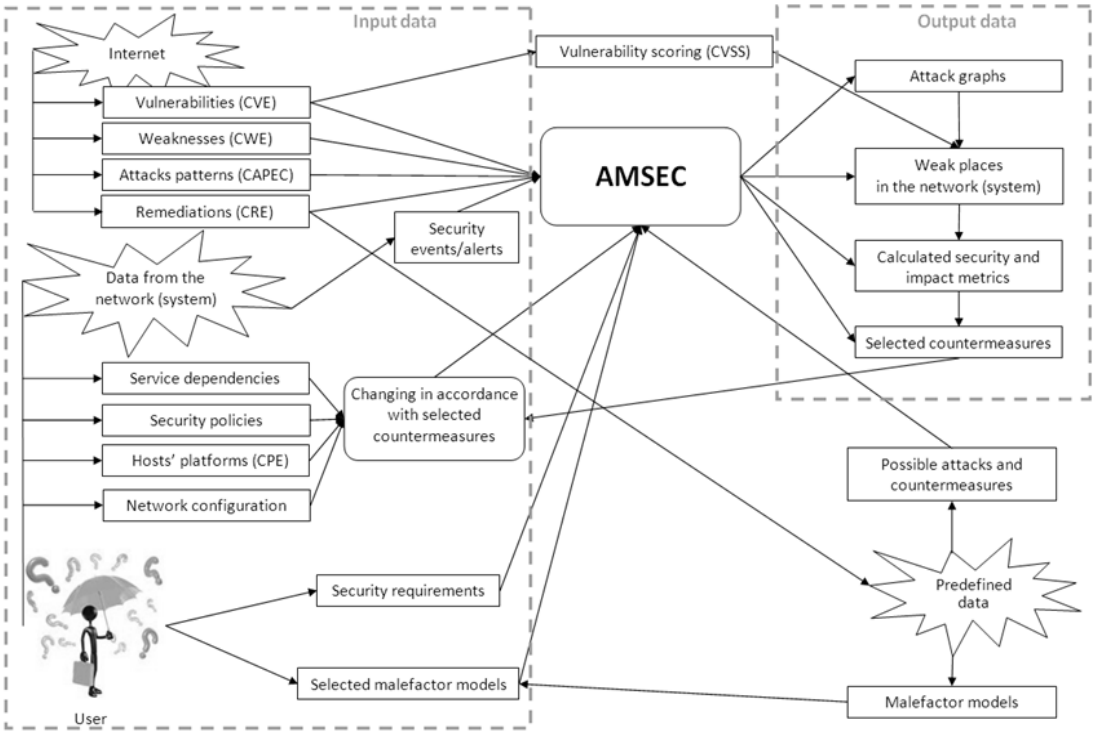

Figure 3. Main data flows in the AMSEC.

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

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

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 the 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) [32], Common Vulnerabilities and Exposures (CVE) [5], Common Attack Pattern Enumeration and Classification (CAPEC) [3], Common Platform Enumeration (CPE) [4], and then translates them into the AMSEC security data repository.

Reports generator shows vulnerabilities detected by the AMSEC, represents "weak" places (hosts or applications responsible for majority of attacks), generates recommendations on strengthening the security level, etc.

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 (Making Security Measurable) related standards [28] 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 for *security repository* consist of two types of data:

(1) data obtained from external sources in the Internet (vulnerabilities, weaknesses, and attack patterns [3] [5] [7] [8]), and

(2) data obtained by analyzing the network (system) and generated by scanning tools and users.

The second type of the input data consists of 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.

The output data are obtained from the AMSEC in the result of modeling and 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. The TVA is used to generate attack graph. This technique is based on enumeration of potential sequences of attack actions (using exploits of known and zero-day vulnerabilities). Two types the analysis are implemented – backward and forward, depending on the place of search initialization (from final or initial nodes) [33] [18].

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, the service dependencies graph is used. This solution allows assessing the impact propagation for the intrusion and response on the basis of CVSS [7] and a set of quantitative metrics [29] [30]. We expand the common model for attack/defense analysis by adding a new object "Service" with specific properties that describe trust relationships between network objects. Thus the additional service layer is defined.

The algorithm of attack graph generation, developed earlier [12] – [14], was changed according to the modifications in the attack model. We added 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 [29] [30] 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 the approach suggested the zero-day vulnerabilities are also taken to into account to generate attack graphs. To do this the approach suggested in [17] is modified by adding additional characteristics which define the probability of existence of the zero day vulnerabilities.

The main idea is to automate the process of selection of hosts which are likely to have zero day vulnerabilities (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 the attack graph. In addition to the attack graph generator, based on the approach of [17], a similar approach suggested in [37] is considered. In this model-based approach the resilience of an

information infrastructure against attacks to unknown (zero-day) vulnerabilities is analyzed by determination of a new generic 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, several approaches, which are based on different security metrics, risk analysis and security evaluation techniques, are used.

4. Implementation

By now a prototype of the 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 [32]. 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 specificatio*n (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;
- 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 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.

The technique which is implemented in the AMSEC can be separated on two main stages:

(1) Gathering information and forming the initial data models;
(2) Attack graph building and security evaluation - building a tree of potential attacks and calculating the security metrics characterizing the analyzed network.

Information gathering phase (the first stage) includes the following sub-stages:

(1) preparation of the initial data (for network scanners and other analysis tools);
(2) gathering the information from external sources;
(3) forming of the models specifying the analyzed network and potential malefactors.

Attack graph building and security evaluation phase (the second stage) can also be separated on the sub-stages:

(1) forming the potential attacks trees based on the data collected on the first stage;
(2) analyzing the attack trees and calculating the security metrics (security levels of hosts and overall network, vulnerabilities of the network, the most dangerous vulnerability, etc.);
(3) forming the report.

The scheme, formalizing the technique implemented in the AMSEC, is shown in Figure 4. It includes three columns, where the actions and procedures of the user (operator), the AMSEC and additional components are specified.

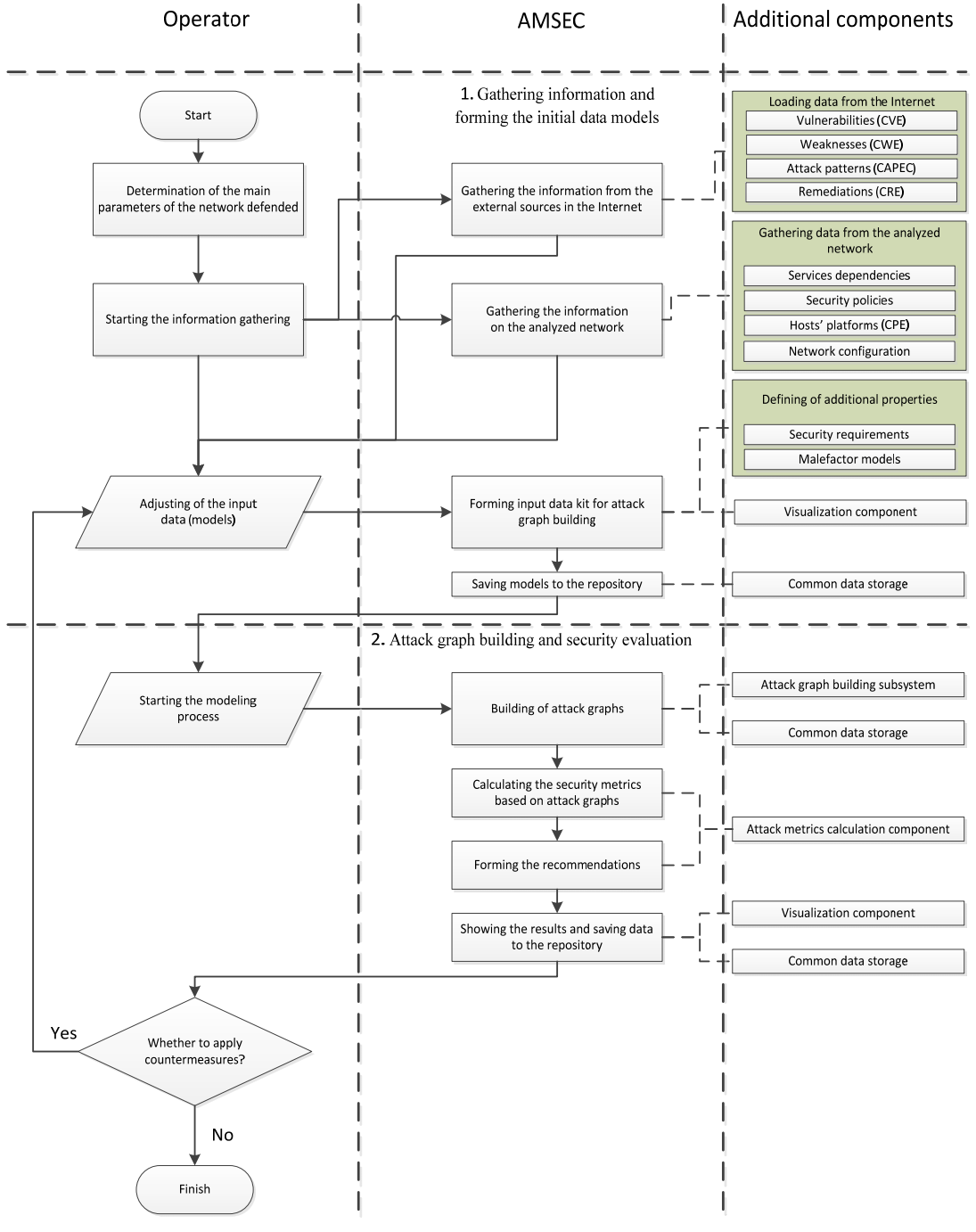

Figure 4. Block scheme of the technique implemented in the AMSEC.

Let us consider *the attack tree building algorithm* in more detail.

The algorithm of generating the common attack graph is based on the attack scenarios model suggested. It is intended to create an attack graph which describes the possible routes of attack actions in the view of malefactor's initial position, skill level, network configuration and used security policy.

The algorithm of generating the common attack graph is based on realization of the following action sequence:

(i) actions which are intended for malefactor's movement from one host onto another;
(ii) reconnaissance actions for detection of "live" hosts;
(iii) reconnaissance scenarios for detected hosts;
(iv) attack actions based on vulnerabilities and auxiliary actions.

All objects of attack graph are divided into two groups:

(i) base objects and
(ii) combined objects.

Base objects define the graph vertexes. They are linked to each other by edges for forming different sequences of malefactor's actions.

Base objects together with connections between them are included in the network model which is used for attack graph generation.

Combined objects are built on the basis of linking the elementary objects by arcs. Objects of types "host" and "attack action" are base (elementary) objects.

Set of objects "hosts" includes all hosts discovered and attacked by malefactors.

Set of objects "attack action" contains all distinguishable actions of malefactors.

All attack actions are divided into the following classes:

(i) Reconnaissance actions;
(ii) Preparatory actions (within the limits of malefactor's privileges). These actions are used for creation of conditions needed to realize other attack actions;
(iii) Actions to gain the privileges of local user and of administrator;
(iv) Confidentiality, Integrity and Availability violation.

Objects of types "route", "threat" and "graph" are combined objects.

Route is a collection of linked vertexes of general attack graph (hosts and attack actions), first of which represents a host (initial malefactor's position) and last has no outgoing arcs.

Threat is a set of various attack routes having identical initial and final vertexes.

We classify threats as follows:

(i) Primary threats – threats of confidentiality, integrity and availability violation;
(ii) Additional threats – threats of gaining information about host or network, gaining privileges of local user and administrator.

Graph is integration of all threats.

The approach of qualitative express assessment of network security level uses the following metrics as basic ones:

- *Criticality(h)* – criticality level of the host *h*;
- *Severity(a)* – criticality level of the attack action *a*;
- *Mortality(a,h)* – damage level caused by the attack action, taking into account the criticality level of the host;
- *Mortality(S)* and *Mortality(T)* – damage level of the route *S* and the threat *T*;
- *AccessComplexity(a)*, *AccessComplexity(S)*, *AccessComplexity(T)* – "access complexity" of the attack action *a*, the route *S* and the threat *T*;
- *Realization(T)* – admissibility of the threat realization;
- *RiskLevel(T)* – risk level of the threat *T*;
- *SecurityLevel* – general security level of the computer network.

The data scheme, which is used for the AMSEC, contains a set of the tables: *network*, *host*, *objectProperty*, *objectPropertyType*, etc. (Figure 5). The data on the analyzed network and the analysis results are stored in these tables.

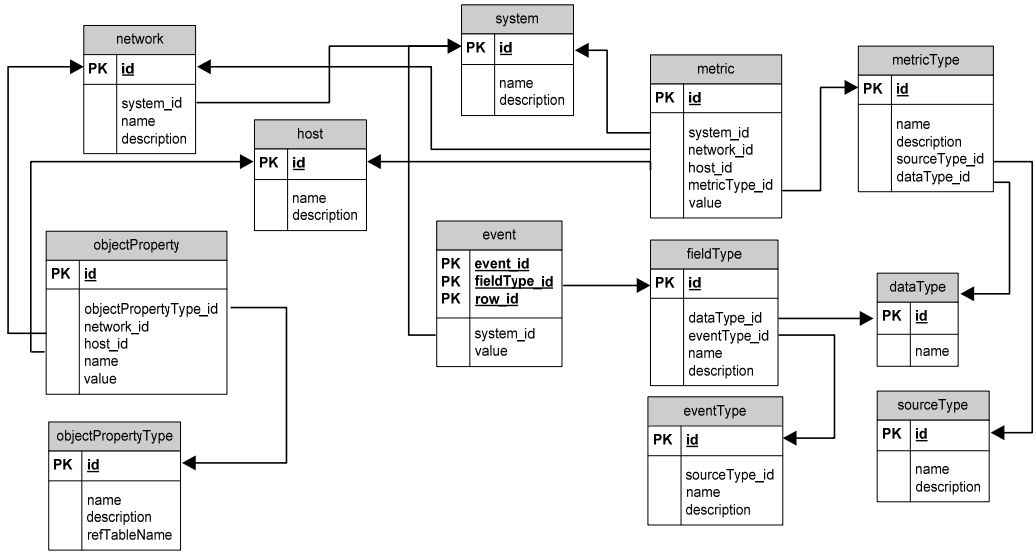

Figure 5. AMSEC storage data scheme.

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

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

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

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

The field *refTableName* can store the name of additional directory table, and the field *value* of the table *objectProperty* contains the identifier of the record from external directory. It is used, for instance, to store the vulnerability list. The additional data scheme for vulnerabilities was created to store NVD database [32]. The table "CVE_List" is the root table which contains all vulnerability identifiers. To make a references between hosts and vulnerabilities, the new property (id=19, name="Vulnerability", refTableName="CVE_List") was created in *objectPropertyType*. Thus, it is possible to add new records to the table *objectProperty* with the type identifier equal to 19, and with the vulnerability record id from the table "CVE_List" in the field *value*.

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

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

5. Experiments

A set of experiments with the AMSEC prototype was conducted. The prototype makes use of scenarios "Critical Infrastructure Process Control (Dam)" and "Managed Enterprise Service Infrastructures" [27]. In the paper the results of security analysis of the dam infrastructure are presented.

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 (supervisory control and data acquisition) components. We can identify three main groups of components in this system: control devices, input/output (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 the malefactor can try to block the dam control commands, fulfill hazardous water release operations and misuse visualization stations. To accommodate all possible attacks let us 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.

Figure 6 illustrates the topology of the tested network and possible attacker's location. Dashed lines separate different groups of computers: External network, Visualization station, Internal network and Control station. Figure 6 shows different control devices (RTU1, RTU2, RTU3 designate Remote Terminal Units, MCU – Monitoring and Control Unit), a gateway, a firewall and computers.

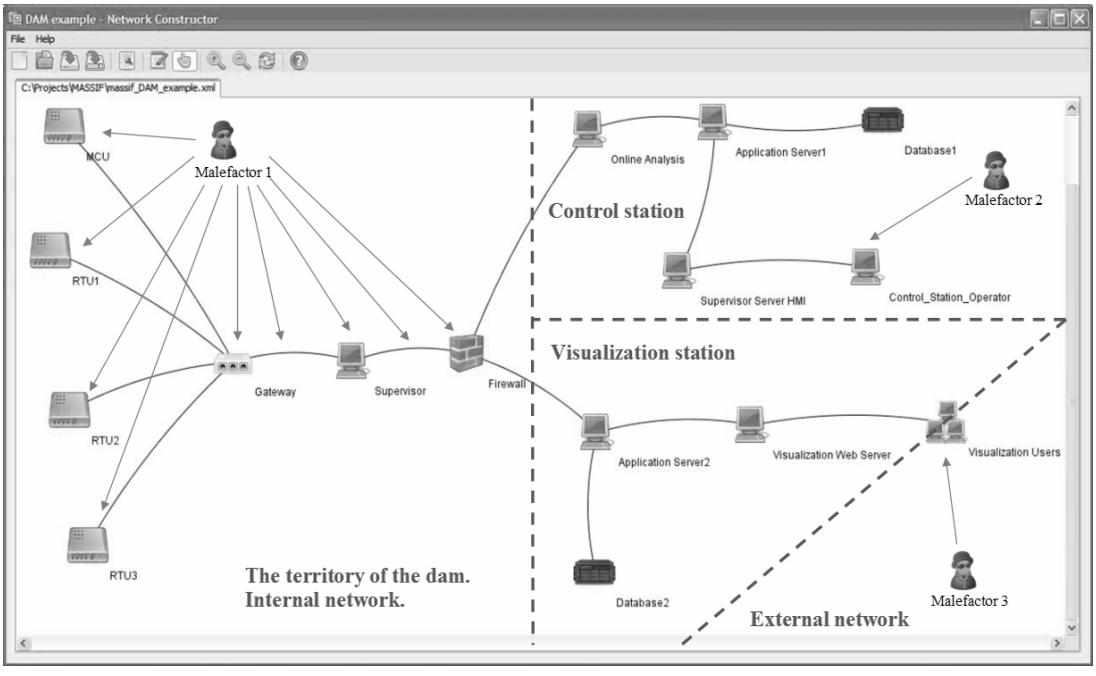

Figure 6. Dam network topology and attacker's locations.

Let us consider the experiments where the Malefactor 3, i.e. the malefactor located outside the controlled network, is chosen as the initiator of attacks. 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, a case with the following software for network hosts is considered: operating system (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.

Figure 7 illustrates different attacks traces that attacker can perform in the tested network.

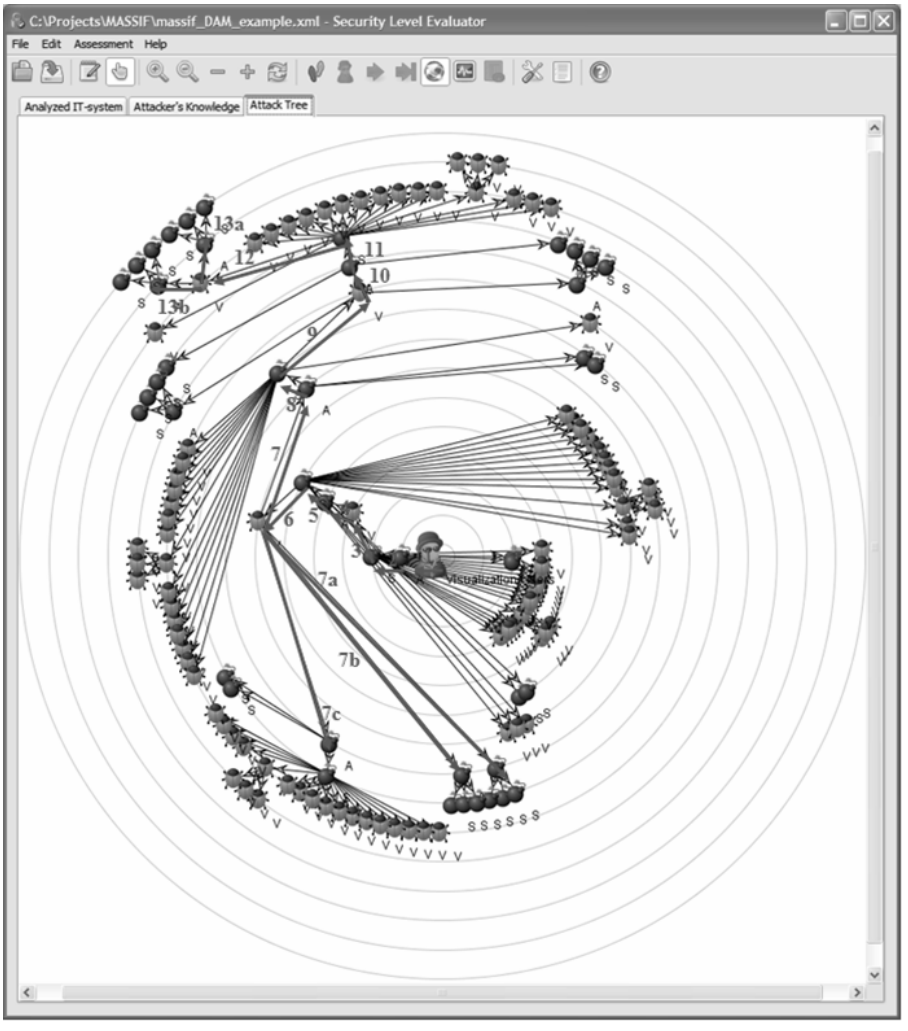

Figure 7. Example of an attack graph.

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" – a 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 [6]. 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.

Let us 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).

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 8 depicts a fragment of the log of the malefactors' actions.

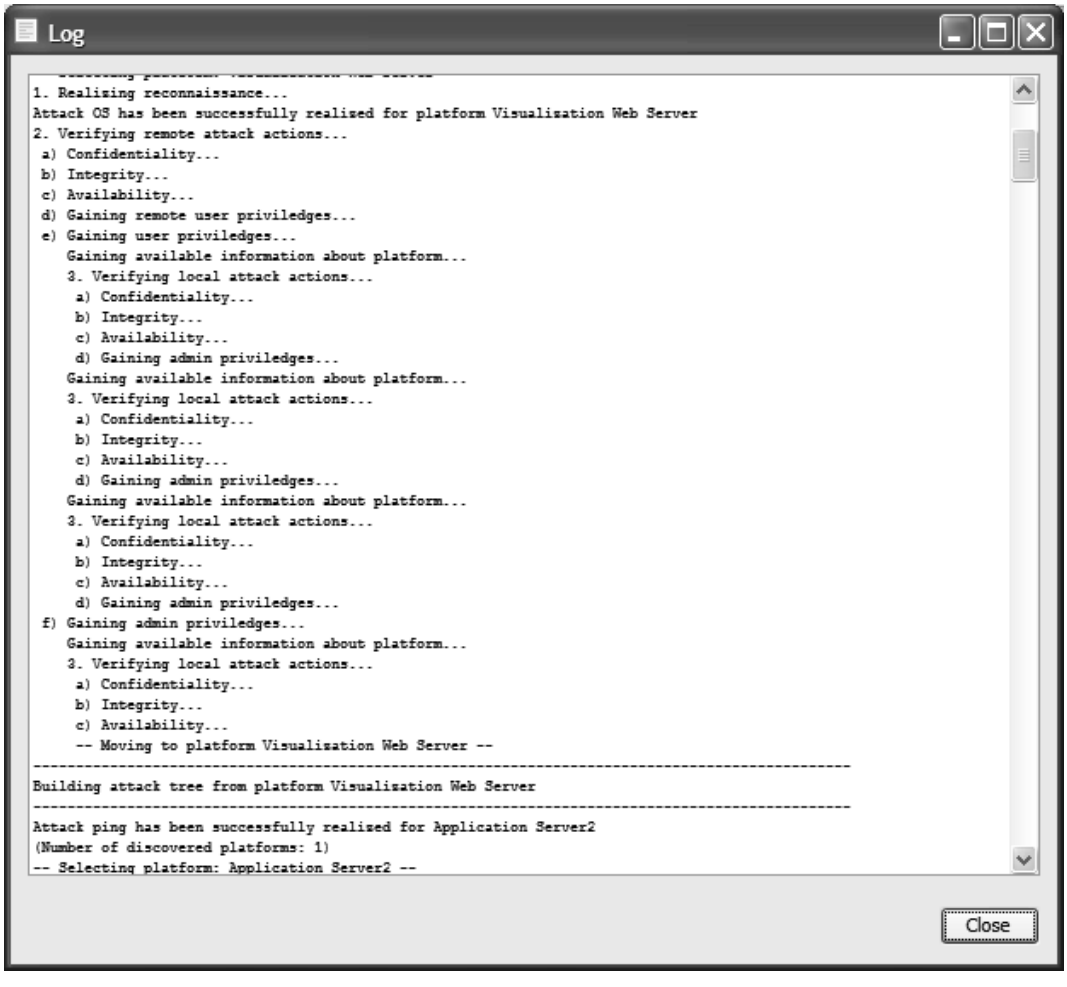

Figure 8. Log of the attack graph building.

This log shows that the malefactor starts to perform attack actions from the host "Visualization Users". This host is a starting point because the malefactor has all privileges in the host according to the specified malefactor model. The selected malefactor is external for the network, and he/she can connect only to the "Visualization Web Server".

Firstly the malefactor gathers the information about the host "Visualization Web Server" and performs attack actions without any privileges on this host.

After several attack actions the malefactor obtains the remote and local users privileges and continues the information gathering.

The final step of the malefactor on this host is to obtain administrators privileges. Then the malefactor scans for accessible hosts and starts new attack actions for a new host. The log shows all attack actions for "Visualization Web Server" and discovering the next host "Application Server2".

Figure 9 shows a fragment of the security evaluation report.

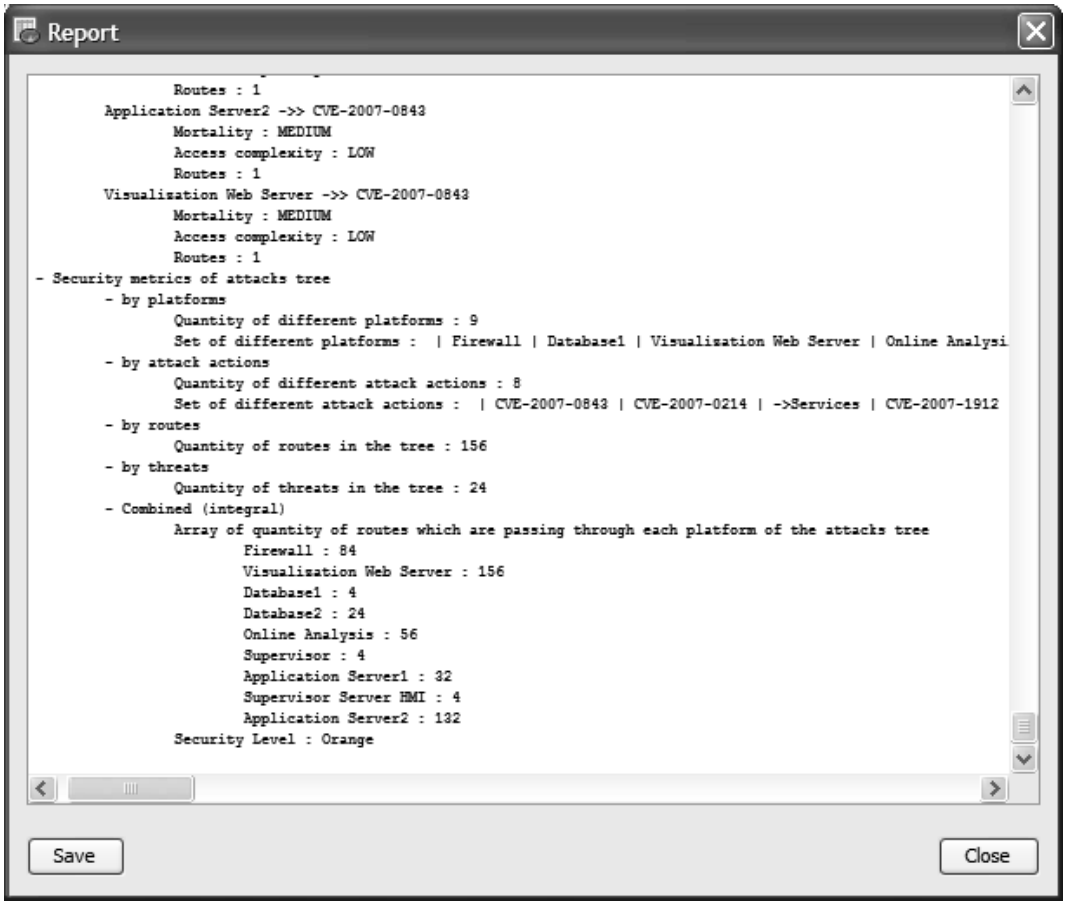

Figure 9. Security evaluation report.

There are 9 hosts in the attack graph. For these hosts eight different successful attack actions were discovered and modeled. The attack graph contains 156 different attack routes. These routes contain 24 security violations (confidentiality, integrity and availability) for different host.

The weak place in the analyzed network for the selected malefactor model is the host "Visualization Web Server" – all 156 routes passed through it. The integrated security level of the network was evaluated as "Orange".

The main recommendation for the system administrator is to increase the protection of the host "Visualization Web Server".

6. Comparison of the AMSEC with Related Systems

In this section, several related systems of different classes, which can fulfill security analysis functions, are analyzed - OpenSKE [24], COMNET III [2] and OPNET [34], Amenaza SecurITree [40], Nessus [31] and Symantec ESM [41]. There are other related systems, but they have similar disadvantages which have been overcome in the AMSEC solutions.

The OpenSKE (Open Security Knowledge Engineered) [24] is an example of the security analysis system which is based on expert system technology. The OpenSKE uses a number of external sources (such as CVE, CPE, OVAL (Open Vulnerability and Assessment Language), CWE (Common Weakness Enumeration), and CAPEC) to create the initial knowledge base. New knowledge is inferred from the existing knowledge by using a set of "if-then" rules. As input the OpenSKE takes information about the system (list of all hosts with user accounts, assets, applications, etc.) and the vulnerabilities (found by security scanner OVAL). The results of the analysis are a list of weaknesses (described by CWE), a list of applicable attack patterns (CAPEC patterns) and a list of compromised assets. The main disadvantage of this system is in the absence of attack graph analysis for specific malefactor. This approach is equivalent to express analysis implemented in the AMSEC, when the security level is calculated without taking into account the network topology. Thus, the accuracy and completeness in detection of potential vulnerabilities and in analyzing attacks is lower.

Stochastic discrete event simulation systems like COMNET III [2] and OPNET [34] allow creating the detailed model of computer networks taking into account the network technologies Frame Relay, TCP/IP, client-server architecture, etc. The results of simulation are the evaluation of network protection against a variety of attacks including resource depletion. During the evaluation, these systems can take into account the information about all applications and hosts sending or receiving network traffic. Disadvantage of these systems is the high resources needed for development. Detailed simulation of the network activity of all services and hosts requires a long time and, therefore, the use of such systems for security analysis is very complicated. In addition, after the changes of network topology and services, it is necessary to fulfill repeated simulation. Thus, taking into account the requirements of operative near real time security analysis, these systems are worse than the AMSEC by efficiency and resource consumption parameters.

Amenaza SecurITree [40] is an example of commercial software which uses attack trees for security analysis. This tool is designed for attack tree building and analyzing, it has a friendly interface and very detailed documentation. The disadvantage of this system in comparison with the AMSEC is the lack of possibility to investigate specific malefactors with his/her capabilities and goals.

Nessus security scanner [31] interacts with the real network and during the scanning cannot penetrate internal network from the external network, if some security system is installed. That is why it usually recognizes only a small number of vulnerabilities. The approach based on malefactor modeling and attack graphs analysis, implemented in the AMSEC, allows detecting all currently known vulnerabilities in the network, regardless of the original location of the malefactor.

In contrast to Nessus, Symantec ESM [41] is a host based scanner, and it is able to detect vulnerabilities on all hosts that have its software module installed. The weak point of this system is the lack of modules for not popular OS. Thus, if one of these operating systems is installed in the analyzed network, Symantec ESM will recognize fewer amounts of vulnerabilities. Unlike Symantec ESM, the AMSEC can use a set of heterogeneous data sources (including Nessus and Symantec ESM). In this way, the amount of recognized vulnerabilities will be much higher.

Let us single out main characteristics which allow comparing the AMSEC solutions with existing systems. All systems considered above allow evaluating the security level of computer networks. These security levels are represented by a set of security metrics, and it is a very complicated task to compare these heterogeneous metrics.

It is supposed that the quality of evaluation depends on completeness of input data for analysis. The following basic groups of input data were chosen:

(1) investigated network parameters;

(2) security system parameters;

(3) malefactor parameters;
(4) parameters of possible attacks.

Information concerning the analyzed network includes data on hosts and topology. Hosts' information can have different level of detail - from minimal (just IP address) to maximal (specification of all services for all operating systems).

Security system description in this analysis is limited by filtering and authentication parameters. These parameters enable us to model attack restrictions.

Malefactor parameters contain the access points, where the malefactor starts the attack actions, the level of knowledge about the investigated system, which allows the malefactor to skip some information gathering steps, and the level of knowledge about possible attack actions and known vulnerabilities.

Specification of existing vulnerabilities and attack actions can be downloaded from the Internet. This functionality is also an important factor for comparison of these systems.

The qualitative comparison of the AMSEC with related systems is shown in Table 1.

The following designations are used in the table: "+" means that the system fully supports this kind of input data; "-" – the support is absent; "+/-" – the system can use the data with some limitation or such functionality requires a lot of additional implementation work. For instance, COMNET III does not support many of existing services, but its architecture allows to implement services with required precision (by developing additional software code). Concerning the security system parameters, both COMNET III and SecurITree allow taking into account a wide variety of authentication mechanisms, but their support also requires additional implementation. Nessus, in turn, can detect security mechanisms by direct and indirect features, but it cannot model the security system parameters which do not exist in the current network.

Table 1. Comparison of the AMSEC with related systems.

Characteristic	**AMSEC**	**OpenSKE**	**COMNET III**	**SecurITree**	**Nessus**	**Symantec ESM**
Investigated network parameters						
Host	+	+	+	+	+	+
Routers	+	+	+	+	+	+
Various OS	+	+	+	+	+	+/-
Various services	+	+	+/-	+	+	+
Network topology	+	-	+	+	+	+
Security system parameters						
Filtering	+	-	+	+/-	+	+
Authentication	+	-	+/-	+/-	+/-	+
Malefactor parameters						
Points of access	+	-	+	+	+	-
Initial knowledge about network	+	-	-	+	+	-
Knowledge level	+	-	-	-	-	-
Attack parameters						
Vulnerability database updating	+	+	-	-	+	+
Attack patterns updating	+	+	-	-	+	-

7. Conclusion

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

(i) 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;
(ii) Taking into account diversity of malefactor's positions, intentions and experience levels;
(iii) 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;
(iv) Taking into account not only attack actions which use known vulnerabilities, but zero days attacks and traditional actions of legitimate users and reconnaissance actions;
(v) Possibility of investigating various threats for different network resources;
(vi) Possibility of detection of bottlenecks – weak places (hosts and applications responsible for the most serious attack actions, routes and threats);
(vii) 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;
(viii) Usage of updated databases, for example, NVD, CVE, CAPEC, CPE, CCE. Usage of CVSS and qualitative techniques of risk analysis.

The implemented prototype of the AMSEC was also described. It can generate an attack tree and calculate security metrics for a predefined network. A simple experiment for critical infrastructure process control on an example of the dam was considered. Several existing approaches for attack modeling and security evaluation were outlined, and the advantages of the solutions implemented in the AMSEC were presented.

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.

Acknowledgements

This research is being supported by grant of the Russian Foundation of Basic Research, 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] 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.
[2] CACI Products Company. http://www.caciasl.com/
[3] CAPEC. Common Attack Pattern Enumeration and Classification. http://capec.mitre.org/
[4] CPE. Common Platform Enumeration. http://cpe.mitre.org/
[5] CVE. Common Vulnerabilities and Exposures. http://cve.mitre.org/
[6] CVE-2007-0214, 2007. Vulnerability Summary for CVE-2007-0214. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-0214
[7] CVSS. Common Vulnerability Scoring System. http://www.first.org/cvss/

[8] CWE. Common Weakness Enumeration. http://cwe.mitre.org/
[9] D R Miller, Sh.Harris, A A Harper, S VanDyke, Ch.Black, Security Information and Event Management (SIEM) Implementation, McGraw–Hill Companies, 2011.
[10] E Bursztein, Extending Anticipation Games with Location, Penalty and Timeline, LSV, ENS Cachan, CNRS, INRIA, France, 2008.
[11] 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.
[12] I Kotenko, M Stepashkin, Attack Graph based Evaluation of Network Security, Lecture Notes in Computer Science, Vol. 4237, 2006, pp. 216-227.
[13] 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.
[14] I Kotenko, M Stepashkin, Network Security Evaluation based on Simulation of Malefactor's Behavior, International Conference on Security and Cryptography (SECRYPT-2006), Portugal, 2006.
[15] J Dawkins, C Campbell, J Hale, Modeling network attacks: Extending the attack tree paradigm, Proceedings of the Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Johns Hopkins University, 2002.
[16] J Ryan, D Ryan, Performance metrics for information security risk management, IEEE Security and Privacy, Vol. 6, 2008, pp. 38-44.
[17] K Ingols, M Chu, R Lippmann, S Webster, S Boyer, Modeling modern network attacks and countermeasures using attack graphs, Proceedings of Annual Computer Security Applications Conference (ACSAC '09), Washington, D.C., USA, IEEE Computer Society, 2009.
[18] 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.
[19] 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.
[20] 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.
[21] L Wang, S Noel, S Jajodia, Minimum-cost network hardening using attack graphs, Computer Communications, Vol. 29, 2006.
[22] 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.
[23] 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.
[24] M M Gamal, D Hasan, A F Hegazy, A Security Analysis Framework Powered by an Expert System, International Journal of Computer Science and Security. Vol. 4, No. 6, 2011, pp. 505–526.
[25] M McQueen, T McQueen, W Boyer, M Chaffin, Empirical estimates and observations of 0-day vulnerabilities, Hawaii International Conference on System Sciences, 2009.
[26] 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.
[27] MASSIF. Massif project. http://www.massif-project.eu
[28] MSM. Making Security Measurable. http://measurablesecurity.mitre.org/ index.html
[29] 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.
[30] 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.

[31] Nessus scanner software, http://www.tenable.com/products/nessus/nessus-product-overview
[32] NVD. National Vulnerability Database. http://nvd.nist.gov/
[33] O Sheyner, J Haines, S Jha, Automated generation and analysis of attack graphs, 2002 IEEE Symposium on Security and Privacy. Berkeley, California, 2002.
[34] OPNET Technologies, Inc. http://www.opnet.com/
[35] R Lippmann, K Ingols, Validating and Restoring Defense in Depth Using Attack Graphs, MILCOM 2006. Washington, DC, 2006.
[36] R P Goldman, A Stochastic Model for Intrusions, Lecture Notes in Computer Science, V.2516. Springer Verlag, 2002, pp. 199-218.
[37] 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.
[38] 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.
[39] S Noel, S Jajodia, B O'Berry, M Jacobs, Efficient minimum-cost network hardening via exploit dependency graphs, ACSAC'03, 2003, P. 86.
[40] SecurITree – Attack graph analysis software. Amenaza Technologies Limited. http://www.amenaza.com/
[41] Symantec Enterprise Security Manager, https://www.symantec.com

Author Bios

Igor Kotenko graduated with honors from St. Petersburg Academy of Space Engineering and St. Petersburg Signal Academy. He obtained the Ph.D. degree (1990) and the National degree of Doctor of Engineering Science (1999). He is Professor of computer science and a head of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences. Author of more than 300 scientific works including 12 books and monographs. His primary research interests include computer network security, artificial intelligence and telecommunications. He was a project leader in the research projects from the US Air Force research department, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. The research performed under these contracts was concerned with innovative methods for network intrusion detection, simulation of network attacks, vulnerability assessment, security protocols, verification and validation of security policy.

Andrey Chechulin received his B.S. and M.S. in Computer science and computer facilities from Saint-Petersburg State Polytechnical University, Saint-Petersburg, Russia. He is now a PhD student at the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences. His primary research interests include computer network security, intrusion detection, analysis of the network traffic and analysis of vulnerabilities.