The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
12-14 September 2013, Berlin, Germany

Computer Attack Modeling and Security Evaluation based on Attack Graphs

Igor Kotenko, Andrey Chechulin
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)
39, 14th Liniya, St. Petersburg, Russia, {ivkote, chechulin}@comsec.spb.ru

Abstract—**The paper considers an approach to computer attack modeling and security evaluation which is suggested to realize in advanced Security Information and Event Management (SIEM) systems. It is based on modeling of malefactors' behavior, building a common attack graph, processing current alerts for real-time adjusting of particular attack graphs, calculating different security metrics and providing security assessment procedures. The approach is intended to be implemented in the framework of the EU MASSIF project. The generalized architecture of the Attack Modeling and Security Evaluation Component (AMSEC), as one of the main analytical components of SIEM systems, is outlined. The main components and techniques for attack modeling and security evaluation are defined. A prototype of the AMSEC is specified. Experiments with this prototype are analyzed. The prototype makes use of the scenario "Managed Enterprise Service Infrastructures".**

Keywords—computer attack modeling; attack graphs; security evaluation; security information and event management

I. INTRODUCTION

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.

Key elements of suggested architectural solutions for attack modeling and security evaluation in SIEM systems 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 [11, 12].

This paper considers the state-of-the-art in attack modeling and security evaluation based on attack graphs, the essence of the approach to analytical attack modeling as well as a generalized architecture of Attack Modeling and Security Evaluation Component (AMSEC) suggested to be developed in the EU MASSIF project [15].

The paper is structured as follows. *Section* II analyzes shortly the main related work. Common framework for computer attack modeling and security assessment is represented in *section* III. Implementation issues are outlined in *section* IV. *Section* V considers experiments carried out. *Conclusion* describes main results of the work and directions of future research.

II. RELATED WORK

There are a lot of papers, which consider different approaches to attack modeling taking into account various classes of attacks.

One of the fist descriptions of attack graphs was suggested by Schneier in 1999 [23]. In this work the approach to manual construction of attack graphs was used for security evaluation. Each graph has a node which represents attack aim and nodes that represents attack actions. In [18] Moore et al. proposed a structured and reusable tree-based form for attacks description and modeling. In 2001, Swiler and Phillips [24] presented one of the first software tools for attack graph generation. Each node of attack graph modeled in this tool represents an attack state and edges specify the attacker's actions.

Lippmann and Ingols [14] consider a tool that is used to construct and analyze automatically attack graphs for detection of firewall configuration defects and host critical vulnerabilities. Information about network vulnerabilities is collected by Nessus security scanner [20], and this information must be manually entered in the database.

Ingols et al. [8] extend this approach to take into account modern network attacks and countermeasures. Particularly, they suggest the improvements to model additional modern threats and countermeasures.

The most common specification of platforms and vulnerabilities is proposed by MITRE Corporation [17]: CPE[3], CVE[4] and CVSS[5].

Common Platform Enumeration (CPE) [3] provides a unified description language for information technology systems, platforms, and packages. It is based on the generic syntax for Uniform Resource Identifiers (URI). CPE contains a formal name format, a language for specifying complex platforms, a method for checking

names against a system, and a description format for binding text and tests to a name.

Common Vulnerabilities and Exposures (CVE) [4] dictionary contains the list of known information security vulnerabilities and exposures. Each vulnerability/exposure has a unique identifier. This enables data exchange between different security products and gives an opportunity to evaluate coverage of tools and services.

Common Vulnerability Scoring System (CVSS) [5] is an open and standardized vulnerability scoring system. CVSS gives an opportunity to prioritize and coordinate a reasonable response to security vulnerabilities via the base, temporal and environmental properties of vulnerability.

Usage of the National Vulnerability Database (NVD) [19] based on CVE dictionary is the basis for constructing of attack graph via known vulnerabilities.

Kheir et al. [10] propose to extend the use of CVSS metrics in the context of intrusion response, by supplying dynamic information about system configuration and service dependencies structured within dependency graphs.

Security metrics are an important element of the security evaluation system. From the system security level point of view a set of security metrics can be outlined: integral metrics of the common security level of system, metrics that define topological characteristics, malefactor characteristics and attack characteristics [1, 6, 7, 9, 22, 25].

The analysis of network security against unknown zero day attacks is also an important topic of research [2, 16, 26, 27].

III. Common Framework

The Attack Modeling and Security Evaluation Component (AMSEC) is intended to complement the SIEM analysis functionality with the capability of attack modeling and security evaluation [11-13].

The *main inputs* for AMSEC are:

- configuration of the computer network (system);
- security policy determining a set of permissions or policy rules;
- event and alerts;
- external databases (DBs) of vulnerabilities, attacks, platform, etc.;
- possible malefactor profiles (as a set of malefactor characteristics);
- required values of security metrics (as a set of requirements to security).

The *main results* of AMSEC are as follows:

- vulnerabilities detected;
- possible routes (graphs) of attacks and attack goals;
- payload internal dependencies;
- bottlenecks ("weak places") in network security;
- preliminary attack trees;
- adjusted attack trees based on changes in the network and alerts;
- predictions of the malefactor's next steps taking into account the current situation;
- security metrics, which can be used for general security level evaluation of computer network (system) and its components;
- attack and countermeasures impacts;
- guidelines for increasing the security level and solutions based on security measures/policies/tools.

The *general architecture* of the AMSEC and its interaction with other components of SIEM system are shown in Figure 1.

Connections, depicted in the figure, show the direction of interactions between different components.

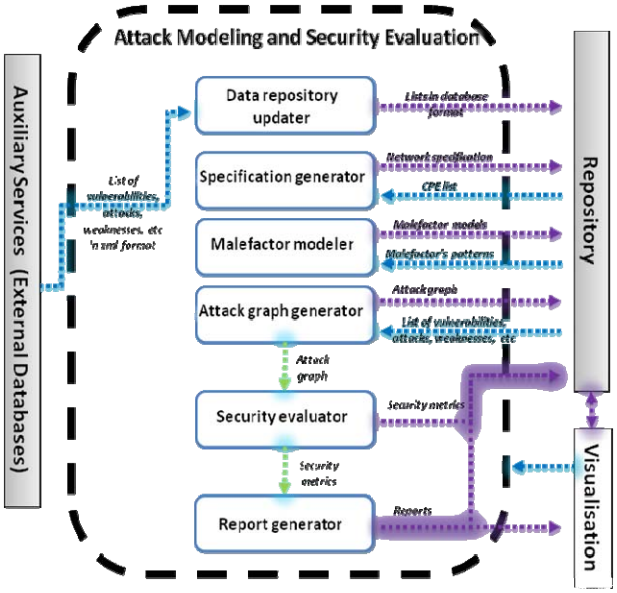


Figure 1. General architecture of AMSEC

Data repository updater downloads the open databases of vulnerabilities, attacks, configurations, weaknesses, platforms, and countermeasures from the external environment (sending requests to external databases for updates and communicating with data sources).

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

Malefactor modeler determines malefactors' individual characteristics, skill level, their initial position (insider/outsider, available points of entry, etc.), the set of permissions, possible actions/attacks already fulfilled (which can be predicted according to events and alerts) and knowledge about the analyzed network.

It also recognizes the most probable malefactor model based on detected attacks and modifies the attack graph based on the changes of the network.

Attack graph generator builds attack graphs (trees) by modeling sequences of malefactor's attack actions in the analyzed computer network using information about available attack actions of different types, services dependencies, network configuration and used security policy. Attack graph generator can also build attack traces taking into account zero-day vulnerabilities – unknown vulnerabilities which are required to compromise network assets.

Security evaluator assists the selection of solutions (validated events and alerts, possible future security events, countermeasures) needed for other SIEM components. It simulates stochastically multi-step attacks and studies the cost and effect of various countermeasures. For example, it generates combined objects and calculates their security metrics in order to evaluate the common security level and possibly make recommendations on strengthening it.

Reports generator shows vulnerabilities detected by the AMSEC, represents "weak" places, generates recommendations on strengthening the security level and depicts other relevant security information.

The AMSEC operates in two main modes (Figure 2):

(1) Design time (or configuration), where the AMSEC is used for design and initial analysis of the network analyzed (or the system under protection). It is a non real-time mode;

(2) Exploitation, where the AMSEC is used for real-time or near real-time operation of the SIEM system.

The functionality of the AMSEC requires the presence of the vulnerability database loaded from the Internet as well as repository which will store the input and output data of the AMSEC. As external repository for the AMSEC the Common Repository is used.

The second component with which AMSEC has tightly integration is the Visualization Component.

Let us consider the main procedures of AMSEC and the interaction of the AMSEC with SIEM components on main modes.

Design (configuration) stage. The AMSEC needs to have a detailed description of protected network topology and configuration for correct and efficient operation. This information is retrieved from the user (through the Visualization system), from predefined data (through Repository) and from sensors placed in the network. As a result, the AMSEC produces attack graphs and calculates security metrics.

Attack graphs can be used to refine event processing rules, and security metrics can be used for decision support and reaction to form the list of recommendations to increase the security level. Since at this stage real-time mode is not required, the information flow can go through the Repository.

Exploitation stage. There are several tasks performed by the AMSEC at this stage: attack graphs adjustment; attack detection improvement by searching matches between real-time events and attack graphs; security metrics evaluation and prediction of potential threats and attacks.

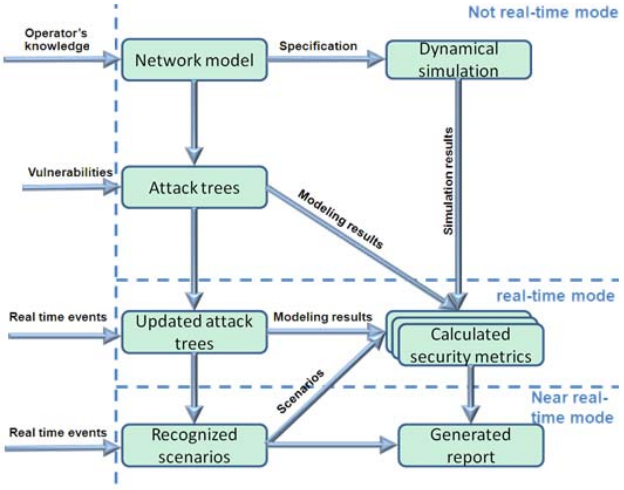


Figure 2. Main procedures of AMSEC

IV. IMPLEMENTATION

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

(1) Gathering information and forming the initial data models;

(2) Attack graph building and security evaluation;

(3) Event analysis.

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 the models specifying the analyzed network and potential malefactors.

At the second stage a 3-dimension matrix is formed for each host according to the following information:

(1) class of attacks (data gathering, preparation activities, escalation of privileges, attack goal realization);

(2) needed access type (remote source without access rights, remote user of the system, local user of the system, administrator);

(3) restriction for malefactors (by malefactor knowledge, zero-day vulnerabilities, etc.).

As a result for each host a set of corteges (attack action class, access type, and malefactor knowledge level) is formed. For each cortege in its turn a list of particular attacks and vulnerabilities needed for these attacks implementation is generated. The total list of vulnerabilities is formed on the base of host software and

hardware description using CPE and public vulnerability databases such as NVD.

The next action is a search of vulnerable software. The examples of patterns used to describe malefactor actions are as follows: CAPEC-310 (Scanning for Vulnerable Software), CAPEC-311 (Fingerprinting Remote Operating Systems), CAPEC-300 (Port Scanning), etc.

On the third stage of attack graph generation, both particular vulnerabilities from the CVE dictionary and patterns like CAPEC-233 (Privilege Escalation) are used.

After forming matrixes of potential attacks, for each host the possible malefactor type and his/her initial location are chosen for the analyzed network.

The examples of the malefactor type are as follows:

(1) External hacker, a user having significant knowledge in information security field, but lacking any direct possibility to connect to the internal network; possible intrusion points, for example, are servers which can be accessed via the Internet (web servers, mail servers, etc.);

(2) Internal user, a user having basic knowledge in information security field with local user or administrator rights;

(3) Worm/virus/botnet, a program that can use a set of vulnerabilities specified in advance. It is supposed that in this case a part of the internal network can be already infected.

The full malefactor model includes the following parameters:

- type (internal, external, complex);
- initial privileges for each host of the network (none, remote user, local user, administrator);
- possible access points in the network;
- knowledge level (defines possible attack actions).

Further for each chosen malefactor model a list of possible goals is generated. For example, for the internal user it could be a revenge (causing maximum damage for the company). The goal of the external hacker could be the access to confidential information located on a server inside the network. For a worm at the first stage a goal can be its distribution, while at the second one it could be carrying out DDoS attacks.

The key elements of the suggested approach are as follows:

(1) for all malefactor models the attack graph is formed in the same time on the basis of information gathered;

(2) for each malefactor model the security metrics are evaluated;

(3) for each malefactor who can successfully realize attacks, the list of possible countermeasures is formed.

Due to the fact that the attack modeling cannot be often fulfilled in real-time, its usage in real-time processes is limited. However, the generated attack graphs keep their actuality for a certain period of time (until significant changes in the security policy or physical network topology occur).

Thanks to this in the frame of the general system of event analysis it is suggested to use the attack graphs constructed in advance. These attack graphs can be used when solving two main tasks:

(1) predicting subsequent malefactor actions and

(2) analysis and detection of their past actions which led the system into its current state.

However, in some cases the attack modeling system needs to update attack graphs. For example, this necessity occurs when host characteristics (software and hardware, criticality, etc.), network topology and a list of possible malefactors are changed, as in these cases the key objects (malefactor models, matrixes of host properties, etc.) are changed.

However in this case attack graphs are updated partly as the changes are calculated only for particular elements of matrixes. Due to this fact the computational complexity of the update decreases significantly.

The AMSEC visualization subsystem provides visual tools for input data configuring and presenting results of attack modeling and security evaluation.

Let us represent the Network Constructor dashboard of the AMSEC used to setup initial data (Figure 3). It is divided into four subviews [21].

The main *view C* shows the topology of the studied network, while the *view A* reflects the hierarchical structure of the network, depicting domains or specified network zones. The graph based techniques are used to represent network topology. Each network object is represented by an icon. The user has possibility to define icons for each type of the network objects. The background color of the icon is used to encode values of the security metrics calculated for the given host, such as Criticality, Mortality, Risk Level.

These metrics are chosen by the user from the predefined list. The brief information about each host is available via a tool tip which appears when mouse hovers over the network object.

The user can configure each host and network using the property *view B*. It can specify predefined properties of the host such as IP address, host type (web server, ftp server, database server, router, firewall, etc.), installed software and hardware, user-defined host criticality. These properties are necessary for attack graph generation. There is also a possibility to define user properties. This property view is updated whenever a particular state node is selected.

The *view D* shows the security metrics calculated for the network itself. As these metrics can have value from the predefined set of values {Low, Medium, Above Medium, High, Undefined}, they are presented in a form of the semaphore signal.

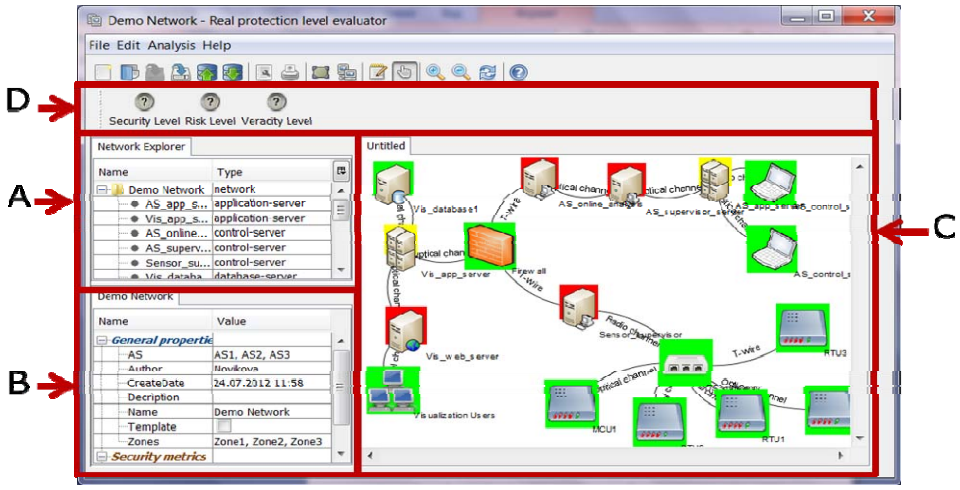


Figure 3. Network Constructor dashboard

To depict the attack modelling results, we use graph based attack representation [21]. Each node of the graph denotes to specific attack action, and their order reflects the sequence of the malefactor actions: the nodes located on one level characterize actions that can be implemented simultaneously or independently from each other, while nodes located on different levels describe actions that are implemented in certain order.

We propose using two graph layouts: tree and radial. Radial layout (Figure 4) is more compact and allows user to view the whole graph.

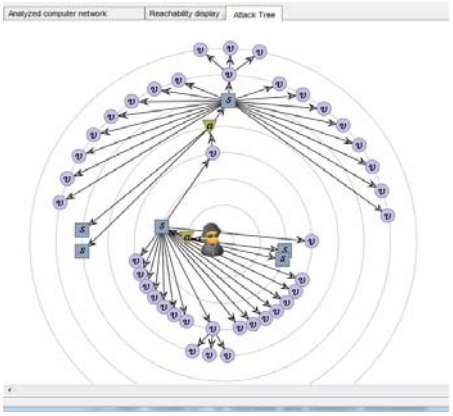


Figure 4. Attack graph displayed using radial layout

This view could be useful when using color encoding of the security metrics of the attack actions, providing general impression on the attack complexity or severity.

The tree view is more traditional and convenient when identifying the sequence of the malefactor actions.

V. Case Study and Experiments

We performed several experiments with the prototype implemented to show the advantages of the proposed framework.

The network for the case study "Managed Enterprise Service Infrastructures" [15] was selected.

The attack modeling and security evaluation process contained the following steps:

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

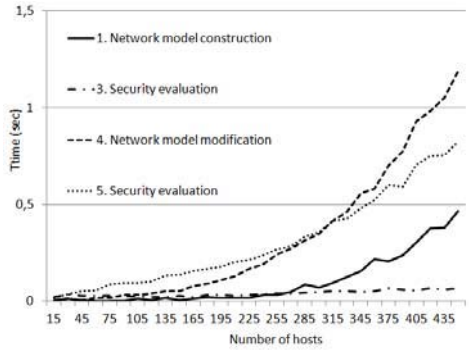
Figure 5 shows the dependency between the time required for different steps of attack modeling and security evaluation process and the amount of hosts in the network. The source network was generated randomly with condition that each host should contain at least one vulnerable software.

Conclusion

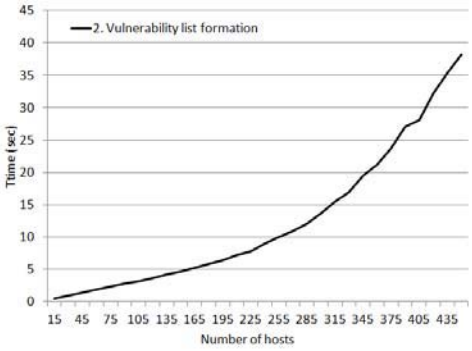
In the paper we presented the framework for computer attack modeling and security assessment component (AMSEC). It outlines also the current prototype of the AMSEC on the whole and the implementation of the techniques for particular attack modeling and security analysis mechanisms. The AMSEC prototype was evaluated by several examples and AMSEC successfully calculated the security metrics for them. All elements of attack modeling and security evaluation described in the paper will be considerably extended and detailed in the further research.

Acknowledgment

This research is being supported by grant of the Russian Foundation of Basic Research (project #13-01-



(a)



(b)

Figure 5. Dependency between the time (sec) of AMSEC functioning and the amount of hosts in the network

00843-a), Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (contract #2.2), State contract #11.519.11.4008 and partly funded by the EU as part of the SecFutur and MASSIF projects.

REFERENCES

[1] B. A. Blakely, "Cyberprints identifying cyber attackers by feature analysis," *Doctoral Dissertation: Iowa State University*, 2012.

[2] E. Bursztein, "Extending Anticipation Games with Location, Penalty and Timeline," *LSV, ENS Cachan, CNRS, INRIA*, France, 2008.

[3] Common Platform Enumeration (CPE). http://cpe.mitre.org/

[4] Common Vulnerabilities and Exposures (CVE). http://cve.mitre.org/

[5] Common Vulnerability Scoring System (CVSS). http://www.first.org/cvss/

[6] R. Dantu, P. Kolan, and J. Cangussu, "Network risk management using attacker profiling," *Security and Communication Networks*, vol.2, No.1, 2009, pp.83–96.

[7] K. J. S. Hoo, "How much is enough? A risk-management approach to computer security," *PhD thesis*, Stanford University, CA, 2000.

[8] K. Ingols, M. Chu, R. Lippmann, S. Webster, and S. Boyer. "Modeling modern network attacks and countermeasures using attack graphs," *Proceedings of the 2009 Annual Computer Security Applications Conference (ACSAC '09)*, Washington, D.C., USA, IEEE Computer Society, 2009. pp.117-126.

[9] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, J. Araujo, "Automated reaction based on risk analysis and attackers skills in intrusion detection systems," *Proceedings of the third International Conference on Risks and Security of Internet and Systems (CRiSIS'08)*, Toezer, Tunisia, 2008, pp.117–124.

[10] N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, and 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.

[11] I. Kotenko, A. Chechulin, and E. Novikova, "Attack Modelling and Security Evaluation for Security Information and Event Management," *Proceedings of the International Conference on Security and Cryptography (SECRYPT 2012)*. Rome, Italy. 24-27 July 2012. pp.391-394.

[12] I. Kotenko, A. Chechulin, "Common Framework for Attack Modeling and Security Evaluation in SIEM Systems," *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing*. Besançon, France, November 20-23, 2012. Los Alamitos, California. IEEE Computer Society. 2012. pp.94-101.

[13] I. Kotenko, A. Chechulin, "Attack Modeling and Security Evaluation in SIEM Systems," *International Transactions on Systems Science and Applications*, Vol.8, December 2012, pp.129-147.

[14] R. Lippmann, and K. Ingols. "Validating and Restoring Defense in Depth Using Attack Graphs," *Proceedings of MILCOM 2006*, Washington, DC, 2006.

[15] MASSIF, 2013. Massif project, http://www.massif-project.eu

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

[17] MITRE Corporation. http://mitre.org/

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

[19] National Vulnerability Database (NVD). http://nvd.nist.gov/

[20] Nessus scanner software. http://www.tenable.com/products/nessus

[21] E. Novikova, I. Kotenko, "Analytical Visualization Techniques for Security Information and Event Management," *Proceedings of the 21th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2013)*. Belfast, Northern Ireland. Los Alamitos, California. IEEE Computer Society. 2013. pp.519-525.

[22] T. Olsson, "Assessing security risk to a network using a statistical model of attacker community competence," *Proceedings of the 11th international conference on Information and Communications Security*, 2009, pp.308–324.

[23] B. Schneier, "Attack Trees – Modeling Security Threats," *Dr. Dobbs Journal*, December, 1999.

[24] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-Attack Graph Generation Tool," *Proceedings of the Second DARPA Information Survivability Conference & Exposition (DISCEX II)*, LosAlamitos, California, vol. II, pp. 307-321, 2001.

[25] The Center for Internet Security, *The CIS Security Metrics*, 2009.

[26] L. Wang, A. Singhal, S. Jajodia, and S. Noel, "K-zero day safety: measuring the security risk of networks against unknown attack," *Proceedings of the 15th European conference on Research in computer security (ESORICS'10)*, Springer-Verlag Berlin, Heidelberg, 2010, pp.573–587.

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