2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

Analytical Visualization Techniques for Security Information and Event Management

Evgenia Novikova
Laboratory of Computer Security Problems
St.Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences (SPIIRAS)
Saint-Petersburg, Russia
novikova@comsec.spb.ru

Igor Kotenko
Laboratory of Computer Security Problems
St. Petersburg Institute for Information and Automation
of the Russian Academy of Sciences (SPIIRAS)
Saint-Petersburg, Russia
ivkote@comsec.spb.ru

***Abstract*— The paper proposes the architecture of the visualization component for the Security Information and Event Management (SIEM) system. The SIEM systems help to comprehend large amounts of the security data. Visualization is the essential part of the SIEM systems. The suggested architecture of the visualization component allows incorporating different visualization technologies and extending easily the application functionality. To illustrate the approach, we developed the prototype of the SIEM visualization component. The paper demonstrates the graphical user interface of the attack modeling component. To increase the efficiency of the visualization techniques we applied principles of the human information perception and interaction issues when designing graphical components.**

Keywords-security information visualization; visualization framework; attack graph visualization

I. Introduction

Modern information systems are characterized by enormous volumes of processed data, and visualization has become the essential tool for data analysis.

In contrast to handling textual data, visualization offers more effective way for analyzing data generated on a daily basis, as it helps to identify general trends, relationships among individual data points or anomalies. This is because the human visual system is a powerful and accurate pattern seeker [1].

Besides visualization helps to cope with increasing data volume, as graphical representation can communicate with a large amount of information encoded in different graphic attributes, such as color, form, size, relative location, etc. [1].

Security information and event management (SIEM) systems are relatively new trend in information security [2]. They are designed to provide vision and clarity on the corporate information system as a whole.

The SIEM systems gather data from different security sensors (e.g. firewalls, routers, IDSs) and detect security incidents in real time by correlating input data. Usually the input data is received in textual format (logs), thus visualization component is essential part of the system.

Within the EU FP7 MASSIF project the SIEM system of a new generation is investigated and developed [3]. It is designed to support intelligent, scalable and multi-level/multi-domain security event processing and predictive security monitoring.

Thus, the design of the visualization subsystem that provides the convenient user interface for functionally new modules such as the predictive security analyzer and the attack modeling module is a challenging task.

The aim of this paper is to analyze security visualization techniques and present a visualization framework applicable for SIEM systems.

Our main contribution is the visualization subsystem architecture that allows easy expanding the SIEM system functionality and integrating different visualization technologies.

To illustrate the suggested approach we present the prototype that provides the graphical interface for the attack modeling and security evaluation component [4].

When developing library of graphical elements, we consider both interaction mechanisms and principles of the visual information perception. The combination of these techniques enforces efficiency of the developed system.

The rest of the paper is organized as follows.

Section II discusses the related work on analytical graphical security data representation and on visualization system architectures.

In section III we describe the proposed visualization subsystem architecture.

Section IV presents the description of the prototype of the SIEM visualization component.

Section V analyzes the paper results and provides insight into our future research.

II. Related Work

A. Visual Models For Security Data Analysis

There are a lot of works that consider different visualization techniques used in information security.

At the moment one of the most comprehensive works in the security visualization is [5] by R. Marty.

He defines main tasks for visualization tools – reporting, monitoring and historical analysis and presents the most widely spread visual models used for perimeter monitoring,

1066-6192/12 $26.00 © 2012 IEEE
DOI 10.1109/PDP.2013.84

insider detection and compliance analysis. For example, different histograms, radial and linear charts are very powerful when presenting statistical security information such as the quantity of transferred or accepted packets, the most often used services, the distribution of protocols in the network traffic. They help to identify different infrastructure attacks such as DDoS, network worms, DNS attacks, etc.

To present data related to defender activities and decision support, the following types of representations are used [5-8]: treemaps, graphs, geographical maps.

At the moment these visualization techniques are implemented in the most of existing SIEM systems [6-8].

Information in the SIEM systems is organized using dashboards as they can communicate important information at a glance. Usually they are grouped according to the role they play - strategic, analytical or operational and contain both graphical and textual (tables) data representation.

Such approach allows users to estimate the same information from different points of view and make decisions more precisely. The SIEM systems allow customizing dashboards flexibly to meet specific user requirements in order to increase overall performance.

In scientific papers the more sophisticated visualization techniques are presented. The significant part of researchers is focused on graphic representation of data and relationships between network activity, security sensor output and attacker activity.

K. Lakkaraju et al. [9] and K. Ohno et al. [10] propose to use a scatter plot to monitor information flows between hosts. Such representation allows, for example, tracing network worm spreading.

Y. Hideshima et al. [11] improve this representation by adding the third dimension – a geographical map and, as the result, two types of representation (logical and geographical) are incorporated in one view allowing to trace network attacks not only in time, but also in space.

J. McPherson et al. [12] propose to use the scatter plot to monitor port activity.

To analyze the port activity in the context of information flows between local and global networks, S. Lau [13] proposes a three-dimensional scatter plot.

C.P. Lee et al. [14] suggest using parallel coordinates to analyze firewall logs.

S. Krasser et al. [15] improve this representation by adding the third dimension, making thus possible to monitor packet flow on the IP address and the port level simultaneously.

F. Mansmann et al. [16] introduce an interesting graph-based metaphor. The nodes of the graph that represent hosts are placed according to the protocol distribution in the network traffic. This representation allows discovering anomalies in the behavior of hosts or higher level network entities.

The graph-based techniques are intensively used to present attack graphs [17-24].

S. Noel [18], for example, investigates the problem of reducing the complexity of attack graphs through visual hierarchical aggregation. He proposes to collapse non-overlapping subgraphs of attack graphs to single graph vertices.

The aggregation operation is recursive according to a predefined aggregation hierarchy. This hierarchy establishes at each level the aggregation rules that are based on either common attributes of attack graph elements or attack graph connectivity.

M. Chu et al. [19] and L. Williams et al. [20] use separate treemaps to display the host groups in each subnet, and the hosts within each treemap are grouped based on their reachability, the attacker privilege level and prerequisites.

Users can also analyze the attack graph step by step to show how attackers progress through a network and learn what vulnerabilities or trust relationships allow critical steps [21-23].

An interesting approach to trace attack evolution is suggested by S. Noel and S. Jajodia in [24]. They apply the adjacency matrix clustering to network attack graphs for attack correlation, prediction and hypothesizing and introduce a graphical technique that shows multiple-step attacks by matching rows and columns of the clustered adjacency matrix. This allows the attack impact/responses to be identified and prioritized according to the number of attack steps to victim machines, and allows the attack origins to be determined.

B. *Architecture of Visualization Systems*

In this section we analyze approaches to visualization system design that are based on the service-oriented architecture which is at the moment a popular paradigm to design scalable distributed applications.

Almost all analyzed papers address the problem of scientific visualization that is characterized by utilizing three-dimensional visualization techniques, complicated surfaces and textures.

Therefore it is more resource intensive then security visualization, but the approaches could be applied due to the need to process large volumes of security data, thus the advantages of the proposed approaches could be exploited.

R. Ananthuni et al. [25] suggest a fat client approach within client-server paradigm in which the visualization software and the database always reside on the server. Clients visualize unique data via submission through web browsers or by accessing previously submitted data on remote server.

Wood et al. [26] propose three-layer architecture: a client layer provides the user interface; a stateful web service middleware layer ensures a published interface to the visualization system; and finally, a visualization component layer which provides the core functionality of visualization techniques.

N. Holmberg et al. [27] focus on the possibility to integrate different visualization technologies in web based application. In their paper they give brief overview of the existing 2D- and 3D-visualization technologies.

B. Grettarson et al. [28] address scalability problems existing in web-based interactive network visualization tools. They propose Web-based Interactive Graph Visualizations (WiGis) and demonstrate fast interactive graph animations for up to hundreds of thousands of nodes in a browser through the use of asynchronous data and image transfer.

III. Visualization Subsystem Framework

The analyzed papers are the basis to develop the SIEM visualization subsystem architecture.

The SIEM visualization subsystem should provide to the user the convenient interface to solve the following tasks:

- monitoring of data in real time (network traffic, network services, availability of hosts);
- work with a repository of events (the historical analysis, formation of reports);
- creation and editing of operation rules for the modules of risk analysis, event correlation, modeling of attacks and countermeasure selection;
- representation of results of attack modeling, risk analysis and countermeasure selection;
- management of security incidents;
- resource management, etc.

The visualization subsystem has to provide a convenient and effective GUI to interact with different functional SIEM components, thus uniting them in one system. Therefore its architecture should allow easy-to-handle functionality extension and provide interaction mechanism between different functional components and user.

The suggested architecture is based on the approach proposed in [26]. We utilize their three-level model but apply it not only to visualization services but also to the SIEM functional services. As it is based on principles of the service oriented architecture, the requirement of functional extensibility of the system is easily fulfilled, and there is no need to redesign and rebuilt an application, when a new component – graphical item or functional component – is added. Besides, service oriented approach conforms to the structural pattern of complex visualization systems "data → visualization → view ↔ control" [29].

The visualization subsystem architecture consists of three layers: (1) *User interface*, (2) *Controlling services middleware* and (3) *Graphical elements*.

The architecture structure is shown in Fig. 1. The arrows reflect information flows between different architecture elements. The separation of the user interface from the other services allows supporting the development of the front-end user forms of different types, beginning from a simple command line and finishing with the rich multi-window interface including various dashboards.

It is supposed that data, which are necessary to visualize, are transferred to the corresponding visualization service which returns the graphical result ready for displaying in application forms.

Such abstraction level makes indistinguishable whether input data are received from the user or from the service and who requested visualization – users or SIEM functional services.

Thus, the controlling services middleware implements interaction between users and other elements of the model. According to the functional payload of the middleware services they could be divided into two groups – the graphical elements controller and the SIEM functional services manager.

The graphical elements controller is responsible for graphical elements management. It provides the standard interface to visualization pipelines: starts and stops visualization pipelines on the request coming from the user interface level or from the SIEM functional service manager. The SIEM functional services manager implements a plug-in mechanism for the services realizing functionality of various SIEM components. Such approach allows developing different functional components independently.

The graphical elements level is a library of necessary graphic primitives – graphs, radar charts, histograms, treemaps, geographical maps, etc. Graphical elements implement mapping of the input data to the visualization models, rendering and user interaction with the input data. Interactivity of the graphical items is an important feature of the visualization tool which helps the user with efficient and quick analysis of large data sets. That is why the principle "overview – filter – details on demand" [30] needs to be considered when developing graphic elements.

The interaction mechanisms should be used in conjunction with specific clustering algorithms that group data according to their properties and connectivity, thus the reduction of the data dimension can be achieved, and therefore the readability of the generated image is increased.

Apart from these techniques different visual effects can be applied in order to improve the perception level of the image. For example, color or blur effect could be very effective when highlighting data sets with similar value of the given property [1].

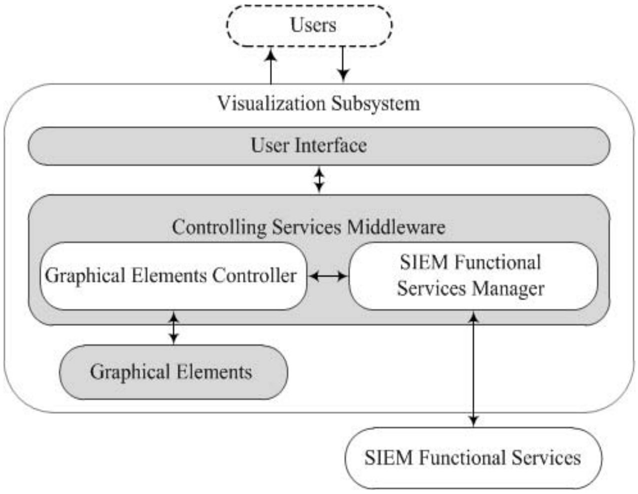
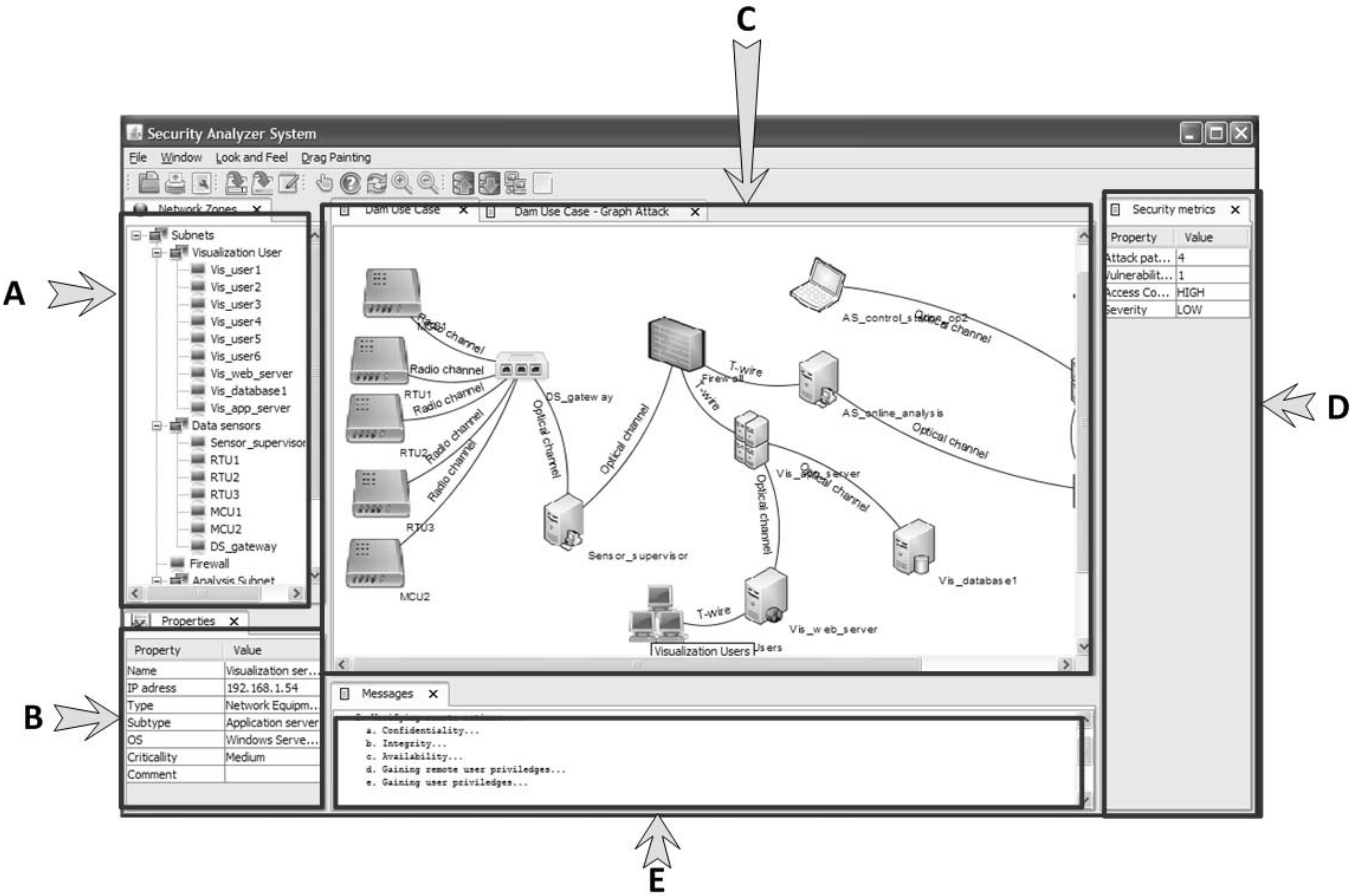

Users
Visualization Subsystem
User Interface
Controlling Services Middleware
Graphical Elements Controller
SIEM Functional Services Manager
Graphical Elements
SIEM Functional Services

Figure 1. Visualization subsystem architecture

The combination of all these techniques helps to avoid cluttering the image with overlapping icons and connected lines - the problem that arises anytime when visualizing large-scale network.

Thus, the offered architecture allows exploiting all advantages of service-oriented approach, including the



Property	Value
Name	Visualization ser…
IP adress	192.168.1.54
Type	Network Equipm…
Subtype	Application server
OS	Windows Serve…
Criticallity	Medium
Comment	

```
a. Confidentiality...
b. Integrity...
c. Availability...
d. Gaining remote user priviledges...
e. Gaining user priviledges...
```

Property	Value
Attack pat…	4
Vulnerabilit…	1
Access Co…	HIGH
Severity	LOW

Figure 2. GUI of the Attack Modelling and Security Evaluation Component

possibility to develop graphical elements effectively, using different visualization technologies (e.g. OpenGL, SVG, Flash).

IV. Visualization Subsystem Implementation

To illustrate the suggested visualization framework, we developed a visualization component prototype.

To realize the plug-in mechanism for SIEM components and graphical elements we used the Apache Felix framework [31] that implements the OSGi technology [32].

We chose the OSGi technology as it facilitates the modular structure of the application and assures the remote management and interoperability of applications and services.

We embedded Apache Felix framework in our application, and it implements functions of the controlling services middleware as it has already defined functions for installing, registering, starting, stopping and uninstalling services such as *install*(), *stop*(), *start*(), *uninstall*().

This framework flexibly solves the problem of the module versioning.

We defined a special interface *GraphicalObject* that provides communication between functional services and graphical elements. This interface describes all basic actions with graphical elements: *create*(*Object data, Canvas canvas*), *setData* (*Object data*), *getControlActions(),*and all graphical services need to implement it.

Let us consider the following example. If a functional service needs graphical presentation of the data (for example, it needs to display the network topology), it tries to obtain available registered services from the Felix framework using specified function *getRegisteredServices*(), the functional service can specify the filter, e.g. "network graph", to get more precise list of services and check properties of selected graphical services to choose the most appropriate one.

If functional service succeeds, then it can simply call the function *create()* of the chosen service, thus creating a new visualization pipeline that presents the network graph.

As a use case we implemented the graphical interface to the Attack Modeling and Security Evaluation Component (AMSEC) of the SIEM system [4].

The visualization subsystem has to provide the interface for configuring AMSEC and present the results of attack modeling and security level assessment (e.g. attack graph, graph of the malefactor knowledge and security level scoring).

The main application view is shown in Fig. 2.

It is divided into five subviews.

The main *view C* shows the topology of the studied network, while *view A* reflects the hierarchical structure of the network, showing domains or specified networks zones. Each icon that reflects the host type can be setup by the user.

The user can configure each host and network using the property *view B*. They 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, host criticality. These properties are necessary for attack graph generation. Also there is a possibility to define user properties.

The visualization system allows displaying initial host information on the *view C* depending on settings defined by the user.

The *view D* shows security metrics computed for each network object including the network itself after network analysis.

We think that such dashboard design gives a general overview about security analysis of the network and communicate a lot of information in a glance.

Thus, the user can analyze calculated host security metrics in the context of initial host configuration; all information is available in different views, but on one dashboard panel.

To depict the attack modeling results we use *graph-based attack representation*. We use Jung [33] library to implement needed graphical elements [33].

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.

The notations used in the attack graph are listed in Table I.

TABLE I. NOTATION USED IN THE ATTACK GRAPH

Notation	Description
👤	initial location of the malefactor
A	specific atomic attack action
S	scenario which does not exploit vulnerabilities
V	attack action that exploits a vulnerability

At the moment we implemented two possible graph layouts: (1) tree layout and (2) radial layout which gives more compact view.

Fig. 3 illustrates different attacks traces that attacker can perform in a tested network using radial layout.

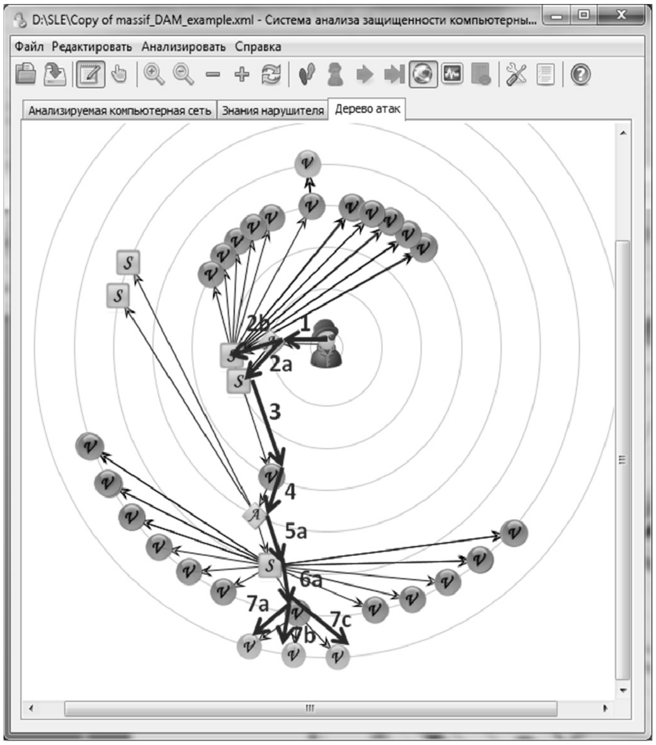
The attacker, carrying out attack actions, is located in the centre of the spherical representation. According to the attack graph the chain of malefactor's actions and their results are as follows:

(1) Detection of nodes connected with the initial malefactor host;

(2) Detection of the software installed on one of the hosts;

(3) Usage of some vulnerability and compromising the host;

(4) Detection of the nodes connected with this host, etc.



D:\SLE\Copy of massif_DAM_example.xml - Система анализа защищенности компьютерны...

Figure 3. Example of an attack graph

The user has a possibility to get detailed information about the malefactor action by clicking on the corresponding node. They are provided by the following data evaluated by the AMSEC: scenario or vulnerability description, severity and the access complexity of the action (Low-Medium-High), host information where the action was implemented.

The brief description of the action, e.g. host name, vulnerability CVE code [34] is also available via tooltip mechanism.

We use color and shape of the node to encode security metrics of the attack graph. We implemented rather traditional color scheme to encode the value of the security metric: green – yellow – orange – red, this is explained by the fact that it is widely used in human everyday life, and red colors are often used to inform about danger while green colors symbolizes norm.

Thus, the user can obtain a general overview about attack complexity or severity.

We also use "black-and-white" effect to emphasize possible ways of the attack spreading. When switched to this mode the user can select the attack action starting from which he (she) wants to follow attack spreading, than all

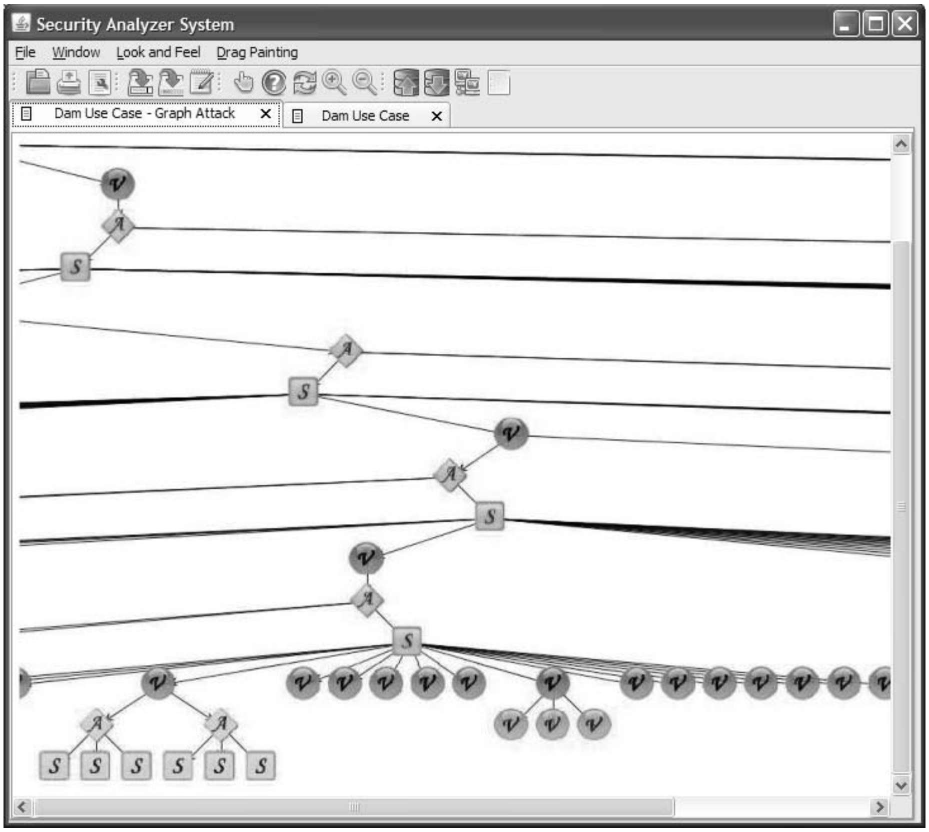


Figure 4. Attack graph with selected attack path

nodes that are not included in the corresponding sequence of the actions are made black-and-white.

This option is shown in Fig. 4. Besides, the user has possibility to hide tree nodes (recursively) by clicking on them.

To present the *graph of the malefactor knowledge,* we implemented two views – graph-based and treemap-based.

In the first view we map the malefactor knowledge on the analyzed network topology. The color is used to encode different characteristics of the compromised hosts evaluated by AMSEC (e.g. host criticality, mortality).

We suppose that this view is common to security officers, thus it could be effectively used. Besides, we adopted a specific force-layout algorithm proposed in [35] that allows grouping hosts in subnets.

Here we also provide the possibility to view detailed information about hosts, e.g. software installed, possible vulnerabilities and misconfigurations.

In the second view we use a treemap to present hierarchical structure of the network. Each subnet is presented by the rectangle, while its nodes are embedded in it. We suppose that this view can be very useful when estimating large-scale hierarchical networks. We use colors to mark compromised nodes. Thus, depending on different initial conditions different graphical attack patterns could be obtained, later these patterns could be applied in real time to predict possible steps of the malefactor.

V. Conclusions

Visualization is very powerful instrument when analyzing large scale data. Efficiency of the visualization tool is determined by used graphical models, interaction mechanisms and perception principles consideration.

In this paper we presented the results of the analysis of the visualization models used in security visualization for the SIEM system, determined the requirements to the SIEM visualization subsystem

We proposed visualization framework based on service-oriented paradigm. It allows easy expanding functionality of the application (both in visualization services and functional services) and incorporating different visualization technologies in one application.

To illustrate our approach, we developed the prototype that offers graphical interface for the Attack Modeling and Security Evaluation component of the SIEM-system.

We considered interaction mechanisms and perception principles when designing graphical elements in order to achieve better understanding of the generated image.

The future work will be devoted to the enhancement of the prototype, performance evaluation of the proposed visualization system and usability assessment of the graphical user interfaces.

ACKNOWLEDGMENT

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

REFERENCES

[1] C. Ware, "Information visualization: Perception for design," 2nd Edition, Elsevier Morgan Kaufman, San Francisco, 2004.

[2] D. Miller, S. Harris, A. Harper, S. VanDyke, "Security Information and Event Management (SIEM) implementation," McGraw-Hill, New York, 2010.

[3] MASSIF Website. http://www.massif-project.eu/

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

[5] R. Marty, "Applied security visualization," Addison Wesley Professional, New York, 2008

[6] OSSIM Website. http://alienvault.com/products/unified-siem/siem

[7] ArcSight Website. http://www.arcsight.com/products/products-esm/

[8] QRadar Website. http://q1labs.com/products/qradar-siem.aspx

[9] K. Lakkaraju, W. Yurcik, A.J. Lee, "NVisionIP: netflow visualizations of system state for security situational awareness," Proc. workshop on visualization and data mining for computer security (VizSEC/DMSEC'04), NY: ACM Press, 2004, pp.65–72.

[10] K. Ohno, H. Koike, K. Koizumi, "IP Matrix: an effective visualization framework for cyber threat monitoring," Proc. 9th International Conference on Information Visualization (IV05), Washington, DC:IEEE Computer Society, 2005, pp.678–685.

[11] Y. Hideshima, H. Koike, "STARMINE: a visualization system for cyber attacks," Proc. Asia Pacific Symposium on Information Visualisation. Darlinghurst: Australian Computer Society, Vol. 60, 2006, pp.131-138.

[12] J. McPherson, K.-L. Ma, P. Krystosk, N. Bartoletti, M. Christensen, "PortVis: A tool for portbased detection of security events," Proc. ACM workshop on Visualization and data mining for computer security (VizSEC/DMSEC '04), NY: ACM Press, 2004, pp.73-81.

[13] S. Lau, "The spinning cube of potential doom," J. Communications of the ACM, Vol. 47(6), 2004, pp. 24-26.

[14] C.P. Lee, J. Trost, N. Gibbs, N. Beyah, J.A. Copeland, "Visual Firewall: Real-time network security monitor," Proc. IEEE Workshop on Visualization for Computer Security (VizSEC'05), Washington, DC: IEEE Computer Society, 2005, pp.129-136.

[15] S. Krasser, G. Conti, J. Grizzard, J. Gribshaw, H. Owen, "Real-time and forensic network data analysis using animated and coordinated visualization," Proc. IEEE Workshop on Information Assurance, NY: IEEE Press, 2005, pp.42-49.

[16] F. Mansmann, L. Meier, D.A. Keim, "Visualization of host behavior for network security," Proc. Workshop on Visualization for Computer Security (VizSEC 2007), Mathematics and Visualization, Springer, Heidelberg, 2008, pp.187-202.

[17] R. Tamassia, B. Palazzi, C. Papamanthou, "Graph drawing for security visualization Graph Drawing," Graph Drawing 2009, LNCS, Vol.5417, I.G. Tolli, M. Patrignani (Eds.), Heidelberg, Springer, 2009, pp.2-13.

[18] S. Noel, "Managing attack graph complexity through visual hierarchical aggregation," Proc. ACM workshop on Visualization and data mining for computer security (VizSEC/DMSEC '04), NY., ACM press, 2004, pp.109-118.

[19] M. Chu, K. Ingols, R. Lippmann, "Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR," Proc. 7th International Symposium on Visualization for Cyber Security (VizSec'10), NY., ACM Press, 2010, pp.22-33.

[20] L. Williams, R. Lippmann, K. Ingols, "An interactive attack graph cascade and reachability display," Proc. ACM workshop on Visualization and data mining for computer security VizSEC/DMSEC'07, Heidelberg, Springer, 2008, pp.221-236.

[21] I. Kotenko, M. Stepashkin, "Analyzing Vulnerabilities and Measuring Security Level at Design and Exploitation Stages of Computer Network Life," Lecture Notes in Computer Science. Springer-Verlag, Vol.3685, 2005, pp.311–324.

[22] I. Kotenko, M. Stepashkin, "Attack Graph based Evaluation of Network Security," The 10th IFIP Conference on Communications and Multimedia Security. CMS'2006. Lecture Notes in Computer Science, Vol. 4237, 2006. pp.216-227.

[23] I. Kotenko, M. Stepashkin, E. Doynikova, "Security Analysis of Computer-aided Systems taking into account Social Engineering Attacks," Proceedings of the 19th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2011). EEE Computer Society, 2011, pp.611-618.

[24] S. Noel, S. Jajodia, "Understanding complex network attack graphs through clustered adjacency matrices," Proc. the 21st Annual Computer Security Applications Conference, 2005, pp.160-169.

[25] R. Ananthuni, B.B. Karki, E.F. Bollig, C.R.S. da Silva, G. Erlebacher, "A web-based visualization and reposition scheme for scientific data," Proc. Int. Conf. on Modeling Simulation and Visualization Methods (MSV'06), CSREA Press, 2006, pp.311-317.

[26] J. Wood, K.W. Brodlie, J. Seo, D.J. Duke, J. Walton, "A web services architecture for visualization," Proc. IEEE 4th International Conference on eScience, Washington, IEEE Computer Society Press, 2008, pp.1-7.

[27] N. Holmberg, B. Wuensche, E. Tempero, "A framework for interactive web-based visualization," 7th Australasian User interface conference (AUIC '06), Vol. 50, Darlinghurst, Australian Computer Society, 2006, pp.137-144.

[28] B. Gretarsson, S. Bostandjievy, J. O'Donovanz, T. Hoellererx, "WiGis: A framework for scalable web-based interactive graph visualizations," Graph Drawing (GD 2009), LNCS, Vol. 5849, D. Eppstein, R. Emden, (Eds.), Heidelberg, Springer, 2009, pp.119-134.

[29] E.H. Chi, "A taxonomy of visualization techniques using the fata state reference model," Proc. IEEE Symposium on Information Visualization, Washington, IEEE Computer Society, 2000, pp 69-75.

[30] B. Shneiderman, "Dynamic queries for visual information seeking," IEEE Software, Vol.11, No.6, 2003, pp.70-77.

[31] Apache Felix framework. http://felix.apache.org/site/index.html

[32] OSGi framework. http://www.osgi.org/Main/HomePage

[33] Jung. http://jung.sourceforge.net

[34] Common vulnerabilities and exposures, http://cve.mitre.org/

[35] T. Itoh, C. Muelder, K.-L. Ma, J. Sese, "A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs," Proc. IEEE Symposium on Pacific Visualization 2009, Washington, IEEE Computer Society, 2009, pp.121-128.