

VisSecAnalyzer: A Visual Analytics Tool for Network Security Assessment

Igor Kotenko, Evgenia Novikova

▶ **To cite this version:**

Igor Kotenko, Evgenia Novikova. VisSecAnalyzer: A Visual Analytics Tool for Network Security Assessment. 1st Cross-Domain Conference and Workshop on Availability, Reliability, and Security in Information Systems (CD-ARES), Sep 2013, Regensburg, Germany. pp.345-360. hal-01506568

HAL Id: hal-01506568

https://inria.hal.science/hal-01506568

Submitted on 12 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

VisSecAnalyzer: a Visual Analytics Tool for Network Security Assessment

Igor Kotenko[1] and Evgenia Novikova[1]

[1] Laboratory of Computer Security Problems
St. Petersburg Institute for Informatics and Automation (SPIIRAS)
39, 14 Liniya, St. Petersburg, Russia
{ivkote, novikova}@comsec.spb.ru

Abstract. Visualization is the essential part of Security Information and Event Management (SIEM) systems. The paper suggests a common framework for SIEM visualization which allows incorporating different visualization technologies and extending easily the application functionality. To illustrate the framework, we developed a SIEM visualization component VisSecAnalyzer. The paper demonstrates its possibilities for the tasks of attack modeling and security assessment. To increase the efficiency of the visualization techniques we applied the principles of the human information perception and interaction.

Keywords: security information visualization, vulnerability analysis and countermeasures, attack graph visualization, information perception and interaction.

1 Introduction

Visual analytics techniques can be efficiently applied when exploring large amounts of data as they can cope with enormous volumes of information and help to extract new knowledge from heterogeneous noisy data. The idea of visual analytics is to combine strengths of human visual system and computational power of automated data processing, making thus possible the development of highly interactive software that allows a user to dive into the data and implement comprehensive analysis in the most promising direction [13].Visual analytics techniques are widely used for security analysis of information systems. However, the most of security visualization tools have been focused largely on active network perimeter monitoring, determining different port scan patterns, detecting anomalies in the "network behavior" of users. Less effort has been done in visualization of the cyber security officer activity, including security assessment, intrusion prevention activity, reasoning and decision support.

In this paper we present visual analytics tool VizSecAnalyzer designed to support the network security level assessment. Its goal is to reveal the most vulnerable nodes of the information system, to form attack patterns, depending on the initial attacker's position and skills, and to adjust countermeasure plan according to the data. It could be used for analyzing software and hardware protection mechanisms used in computer

networks and prevention of possible attacks. As the implementation of the preventive measures decreases the risks of security incidents the usage of the tool can increase the overall efficiency of homeland defense activity. The visualization in VizSecAnalyzer consists of interactive graphs, treemaps and pie charts allowing exploration of large-scale networks. Specifically, the contribution of this paper to the field of security visualization is a common framework and the visualization tool that supports security analysis of network "week" places (vulnerabilities, misconfigurations) in context of possible consequences of their exploitation. The paper is organized as follows. *Section 2* analyzes the visualization techniques used for network security assessment. *Section 3* describes the proposed visual models and interaction techniques implemented in the tool. *Section 4* presents case study for network level assessment and countermeasure plan adjustment. Conclusion analyzes the paper results and provides insight into the future research.

2 Related Work

A lot of work has been done on graphic representation of output of security sensors such as IDS, firewalls, etc. [7, 12, 16 - 18, 27, 30]. However, the visualization of complex security events generated by intrusion detection systems that process heterogeneous security information in large-scale architecture, such as presented for example in [6], is not studied extensively. As our tool is purposed to detect potential weaknesses of the network security measures we focus on the tools designed to support preventive activity of the security officer in this section. There has been much research in visualization of security policies and control resource access. Graph-based techniques are usually used to depict and explore control resource access rules [20, 21]. The graph nodes correspond to users, user groups, and resources. The links between nodes reflect user activities or resources the users accessed. The colors are used to highlight user role. Application of graph layouting techniques based on graph semantics can exposure users with similar behaviors, as well as allows detecting incorrect resource access rules or determining an insider threat.

In [1, 2] a graph-based visualization technique is applied for assessing topology based policies used in social networks. They built a graph where the vertexes correspond to the users' profiles and the edges reflect access permissions between users. The color is used to display reachable and unreachable neighborhood regions of the selected user. The visualization tool allows the user to analyze his/her profile from the view point of another user at his/her neighborhood. By clicking on the vertex of the selected user it is possible to get information about resources accessible for this user and a list of primitives that the selected user can initiate against the profile owner.

Heitzmann et al. [10] suggested the visual representation of access control permissions in a standard hierarchical file system in the form of treemaps. The colors are used to display the permissions. For each file or folder the associated node in the treemap is painted green, red, or gray, if the file's permissions are weaker, stronger, or the same as those specified by the baseline which can take one of the values "no access", "read", "read&write", and "full control". The tool draws an orange border around treemap nodes associated with files or folders where inheritance is broken.

Another group of visualization tools purposed for *assessing network security level* visualizes firewall rules in order to enhance understanding and inspecting them. These tools exploit more sophisticated visual models. For example, Tran et al. [31] developed a tool called PolicyVis which maps firewall rules onto the 2D space in form of the rectangles. It uses three different rule fields to build the policy graph, two of which are used to define the vertical and horizontal coordinates of the rectangle and the third field is integrated into the visualization object. The color is used to encode different kinds of traffic (accepted or denied). Since the colors are set transparently, the overlapping rules can be effectively recognized. Overlapping of the rectangles can notify about a potential anomaly.

Original approach for visual analysis of the firewall rules is suggested by Mansmann et al. [19]. It is based on hierarchical sunburst visualization technique [29], which puts a root node as a circle in the middle of the visualization and then recursively maps the elements of each hierarchy level onto ring segments. To be able to represent graphically firewall rules the authors proposed the following hierarchical structure. The first level after the root node for the rules visualization consists of the names of different access lists as shown in Fig. 1.

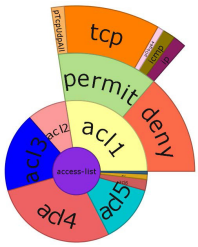


Fig. 1. Starburst visualization of firewall rules [19]

The second level contains the access privileges (“permit” or “deny”), the 3rd one - the protocol, followed by the source and destination dimensions. To make the exploration process easier the authors established the following color scheme. The fixed colors are used for the most common keywords (e.g. “TCP”, “any”, “permit”, “deny”). Less common keywords and names are assigned repeating colors. Besides, the user can change the depth of the visible graph by establishing the depth level number. To improve the visibility of a node and the readability of its label, the width of a segment can be changed interactively using the mouse wheel.

Vulnerability analysis is a critical component in the evaluation of the network security. Currently, the visualization techniques used to display vulnerability scanner reports are limited to treemaps [9, 20]. For example, Nv tool [9] uses treemaps and linked histograms to allow security analysts and systems administrators to analyze vulnerabilities detected by the Nessus vulnerability scanner [23]. Apart from visualization of the Nessus scans, it supports the analysis of sequential scans by showing which vulnerabilities have been fixed, remain open, or are newly discovered. Nv tool uses a semantic based color scheme where, for example, different colors are used for fixed vulnerabilities, new ones, and open vulnerabilities (Fig. 2).

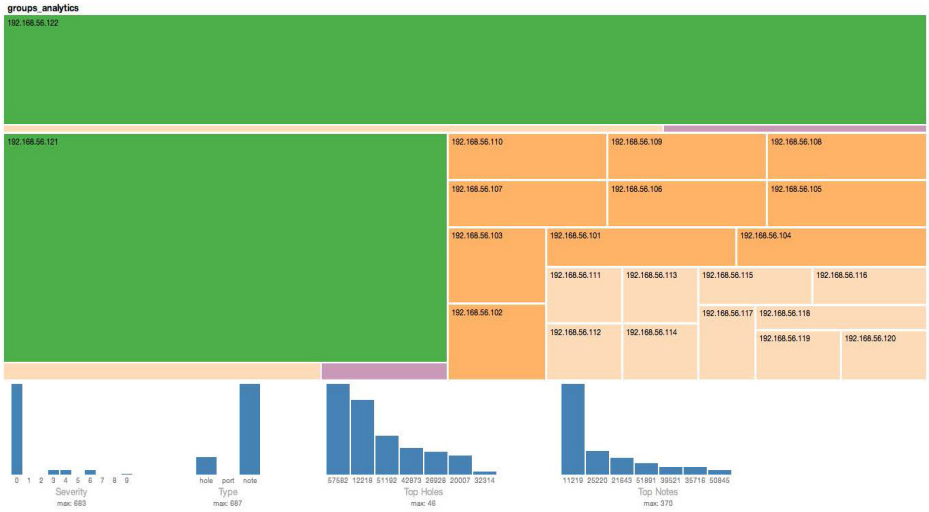


Fig. 2. Using treemaps in the NV tool [9]

Another powerful instrument for security assessment is *attack graph generation*. They are used to determine if designated goal states can be reached by attackers attempting to penetrate computer networks from initial starting states. For this use, they consider the graphs in which the starting node represents an attacker initial position. Other nodes and edges represent actions the attacker takes and changes in the network state caused by these actions. Actions typically involve exploits or exploit steps that take advantage of vulnerabilities in software or protocols.

Natural graphical representation of the attack graphs are graphs themselves [11, 24]. However in the standard exploit dependency representation, the attack graph complexity is $O(scn^2)$, for n machines in the attack graph. Here, s is the average number of exploits against a machine, independent of any particular attacking machine, and factor c is the average number of security conditions per machine.

In order to reduce attack graph complexity several approaches based on attack graph aggregation were suggested. In [25] the graph complexity is achieved through interactive visualization, which includes hierarchical aggregation of graph elements. Aggregation collapses recursively non-overlapping subgraphs of the attack graph to single graph vertices, reducing of attack graph complexity.

In [24, 25] a matrix-based approach is suggested to visualize attack graphs. The attack graph consisting of the n vertices is represented by the $n \times n$ adjacency matrix A, where element a_{ij} of A indicates the presence of an edge from vertex i to vertex j. As in attack graphs, it is possible to have multiple edges between a pair of vertices, for example, multiple exploits between a pair of machines, the authors suggest either to record the actual number of edges, or simply to fix the presence of at least one edge. This visualization technique allows constructing attack patterns by reordering rows and columns of an adjacency matrix as these operations do not affect the structure of the attack graph.

An alternative approach for visualizing the attack graph was proposed in [3,8,32, 33]. It allows mapping the attack graph on the network topology. Separate treemaps are used to represent subnet groups. The inner subgroups are colored according to the selected subnet attribute, and the relative size of each is proportional to the number of hosts it comprises. The attack reachability display is shown in Fig. 3. The user is provided with the possibility to position and resize the subnets to form more intuitive layout. It is also possible to display either incoming or outcoming attack graph edges which are drawn to all nodes that can be reached from the selected node or reach it.

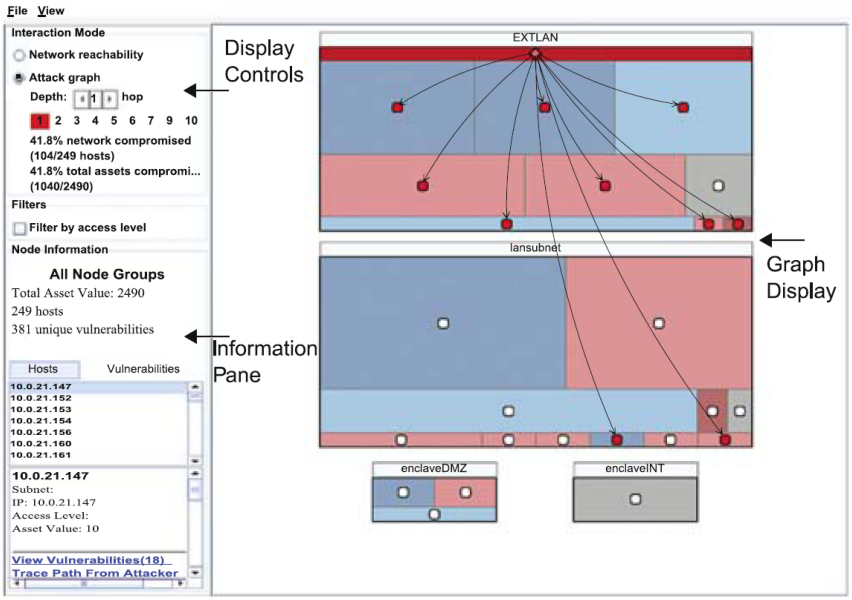


Fig. 3. Attack reachability display [33]

3 VisSecAnalyzer Design

The goal of the VisSecAnalyzer is to provide visual support for cyber security officer when evaluating general network security, analyzing severity of the detected vulnerabilities and assessing efficiency of possible countermeasures such as software update/removal [26].

The VisSecAnalyzer architecture consists of three layers: (1) User interface, (2) Controlling services middleware and (3) Graphical elements. The architecture structure is shown in Fig. 4. 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.

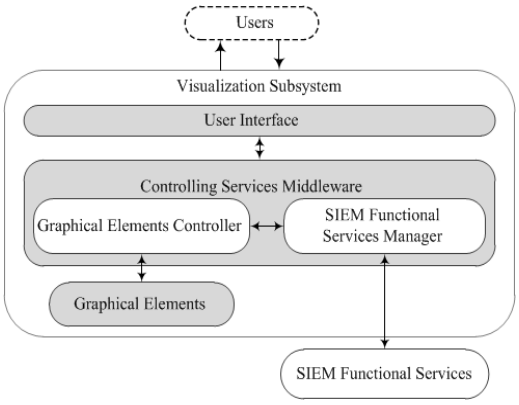


Fig. 4. VisSecAnalyzer architecture

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" 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. The graphical user interface (GUI) of the tool is designed in such way that it stimulates the exploration process of the user and enhances the understanding of the network "week" places origin and the consequences of their exploitation.

The VisSecAnalyzer visualizes the results of the Attack Modeling and Security Evaluation Component (AMSEC) [15] that produces report on security level of the analyzed network. The AMSEC works as one of the SIEM functional services. The architecture used allows plugging components for visualization security events generated by different security sensors such as firewalls and intrusion detection systems.

3.1 Data

The main input data for the AMSEC are network topology, host configuration data including software, hardware, user-defined criticality and veracity level, and network alerts. The AMSEC assesses the specified network producing a set of security metrics associated with the network itself and each host in particular. These metrics include, for example, Security Level, Risk Level, and Veracity Level. It also lists a set of vulnerabilities and exposures formed using the host configuration and a vulnerability database, such as NVD [22] or loaded from vulnerability scanner report if available for the specified network. Each vulnerability is described using CVE-code [4], security score [5], brief description and data how it could be patched or mitigated.

Apart from calculated security metrics the AMSEC outputs the attack graph for a given malefactor model [15]. Each node of the graph denotes a specific attack action, and the attack action order reflects the sequence of 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 attack action is characterized by its action type, access complexity, mortality, severity, vulnerabilities or exposures used, attacking host and target host.

All this information is used when visualizing reports of the AMSEC as it could provide a clear understanding of the security problem existing in the system.

3.2 Visualization and User Interactions

The VisSecAnalyzer GUI consists of several views designed to efficiently support cyber security analysis process. The main window is divided into subviews (Fig. 5).

The main view 1 shows the topology of the network, while view 2 reflects the structure of the network, depicting domains or specified user network groups.

The user can configure each host and network using the Property View 3. It is possible to specify the 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. This property view is updated whenever a particular state node is selected. Thus user always has details at hand.

The view 4 shows the security metrics calculated for the network itself: Security level, Risk Level, and Veracity level. Graphical elements corresponding to these security metrics are located on the main tool bar in order to attract user attention immediately. As these metrics can have value from the predefined set of values {Low, Medium, Above Medium, High, Undefined}, they are presented in as a semaphore.

We suppose that such dashboard design gives a general overview about security state 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.

The graph based techniques are used to represent network topology. Each network object is represented by an icon. The user has the 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 the mouse hovers over the network object.

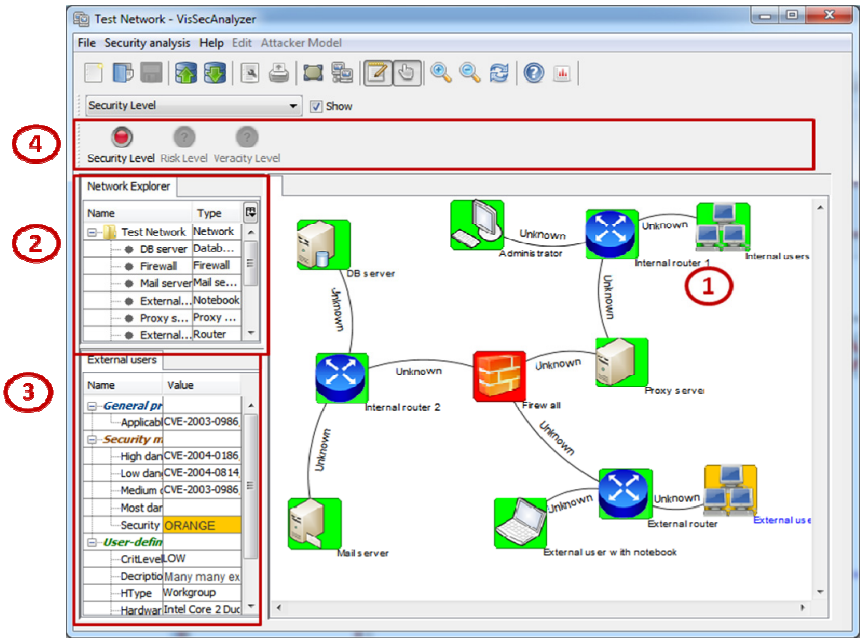


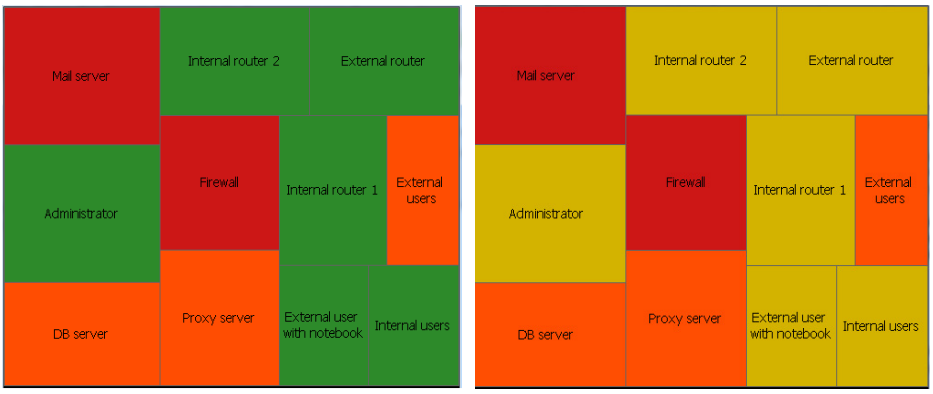
Fig. 5. Main form of the VisSecAnalyzer

In order to display large scale networks a simple geometric zooming and a semantic zooming are used. Using the semantic zooming the nodes can be aggregated according to their properties (belonging to the domain, group, etc.). This aggregation is done interactively - the user can collapse a part of the network or expand aggregated node by choosing corresponding menu item from the context menu of the node.

We use interactive treemaps to present both a vulnerability report and a network security report (Fig. 6). Each nested rectangle displays a network host. The user has an option to choose host attributes (user-defined host criticality, number and severity of detected vulnerabilities and security level) defining rectangle size and color.

By default the size is determined by the user-defined host criticality, and color is defined by security level (Fig. 6a) and vulnerability severity (Fig. 6b). Clicking on the frame of the rectangle denoting the network group (domain or user-defined group) zooms in or zooms out the selected part of the network, while clicking on the rectangle itself updates Property View of the corresponding host.

To display security metrics we use traditional semantic based color scheme, ranging between green and red. When mapping metrics value as green, colors usually notify about normal state, while red ones mean danger. But we avoid using green colors in the reporting about vulnerabilities as they could be confusing to the user. Instead we use yellow color to encode vulnerabilities with low severity level.



(a) (b)

Fig. 6. Security reports in form treemaps: (a) user-defined criticality vs. security level; (b) user-defined criticality vs. vulnerability severity

To depict the attack modeling results, we use graph based attack representation. The notations used to display attack graph are listed in Table 1. We use both color and shape to encode the type of the malefactor action. Such solution allows using color to display different security metrics calculated for each action.

Table 1. Notation used to display elements of attack graph

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

Attack graphs help to investigate attack deployment in the analyzed network, following malefactor actions step by step (Fig. 7). We implemented two possible graph layouts: (1) tree layout (Fig. 7a) and (2) radial layout which gives more compact view (Fig. 7b).

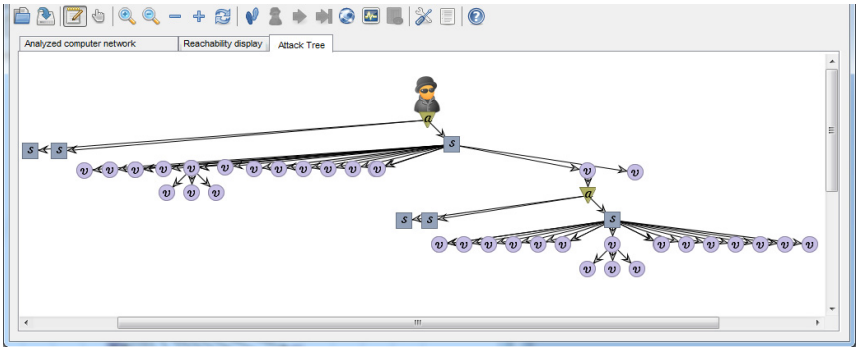
But network attack graphs can be both large and exhibit very dense connectivity making their analysis unfeasible task [24].

In order to solve this problem we propose using the following interaction techniques to make the analysis process easier and usage attack modeling techniques for security tasks more efficient.

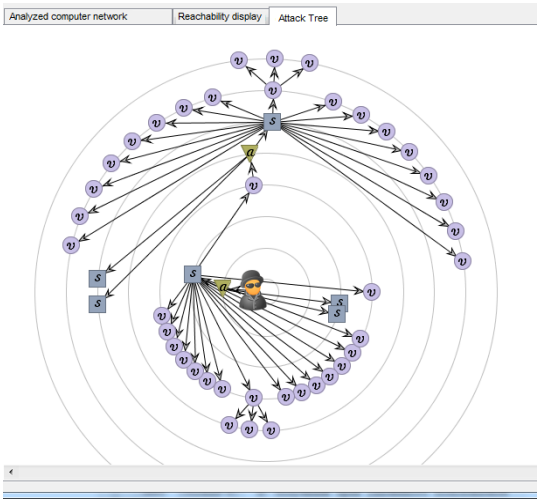
Geometric zooming. It allows user to focus on a specific part of the attack graph and decrease the graph dense connectivity. The distance between graph nodes can be changed interactively using the mouse wheel.

Layout reconfiguration. We propose using two graph layouts: tree and radial. Radial layout is more compact and allows user to view the whole graph.

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 convenient when identifying the sequence of the malefactor actions.



(a)



(b)

Fig. 7. Attack graph representation: (a) tree layout; (b) radial layout

Semantic zooming (aggregation). We suggest applying semantic based aggregation techniques for reducing the complexity of the graph. Depending on such graph node properties as action type, host or node connectivity the graph vertexes could be replaced by one meta-node. The used aggregation rules are depicted in Fig. 8.

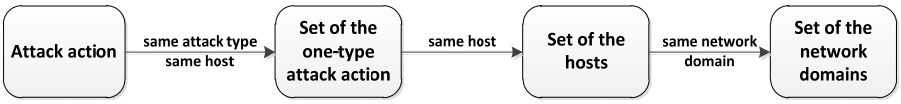


Fig. 8. Attack graph aggregation rules

Details on demand. By clicking on the corresponding graph node the user gets detailed information shown in Property View. This information includes attack type, attacking host and targeted host, user criticality value, vulnerability specific information (metric, CVE code [3] and description), and host metrics if calculated (Mortality, Risk Level). This informational display is updated whenever a particular graph node is selected.

Linking and Brushing. This effect can be applied in order to outline the path of the attack. When switched to this mode the user can select the attack action by clicking on it, this will make all subsequent and precedent nodes linked to the node of his/her interest remain colored while the rest will be drawn in grayscale (Fig. 9).

The graph based attack views are good when studying the sequence of the malefactor actions, but they do not provide an intuitive view on the compromised hosts in the network. We consider using treemaps as they can compactly represent the network. In this case the color of the rectangle reflects the state of the host (red - the host is reached by attacker, green - the attacker cannot get access to the host).

In this case it is possible to use green color as it informs the user that the network hosts are secured and do not need deployment of new protection mechanisms.

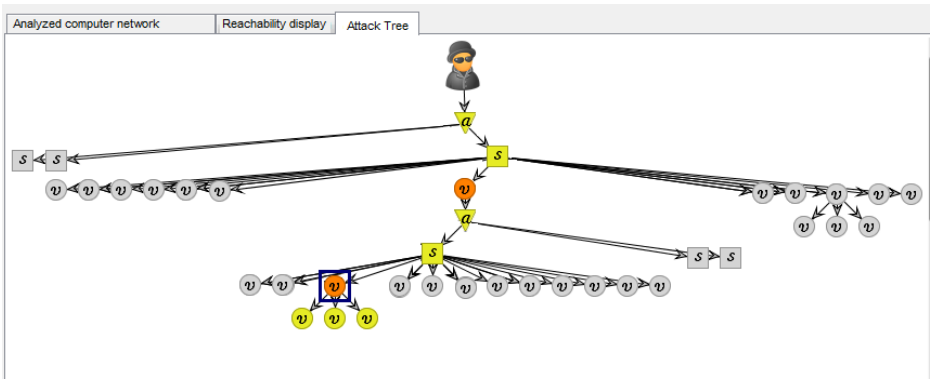


Fig. 9. Effect of linking and brushing

4 Case Study and Usability Evaluation

In order to assess the efficiency of the visualization mechanisms developed we created a test network with typical topology. It consisted of several servers (database server, mail server, internal and external web servers), firewall, IDS and more that 100 workstations (internal users). On all hosts Microsoft SQL Server 3 was installed, DBMS Apache Derby 10.1 was used on database servers, mail server had Microsoft Exchange Server 7 SP1 installed, while workstations were equipped with mail client Microsoft Outlook 2003 and Microsoft Office 2003.

In order to assess security level of the analyzed network the NVD vulnerability databases were used. Then we calculated integrated security level for the network, and it turned to be a red one (very high criticality). The tool provided a possibility to

highlight the security level of nodes in order to determine the most critical ones. In our case the most critical hosts were firewall and mail server.

By clicking on each critical host it was possible to see vulnerabilities grouped according their severity level, thus it is possible to identify the most critical ones at ones.

Thus for the mail server the most dangerous vulnerability turned to be CVE-2007-3898, that allows remote attackers to spoof DNS replies, poison the DNS cache, and facilitate further attack vectors.

Using treemap, the sizes of rectangles of which depend on user-defined criticality level and color is defined by security level, we could see that the mail server should be served in first turn and then firewall in order to increase the security level.

Then we could analyze the consequences of the mail-server compromise. By setting initial location to the mail server and constructing attack graph we could conduct reachability analysis.

Fig. 10 illustrates the compromised and secured hosts presented in the form of treemap. It is clearly seen the malefactor could reach data base server, thus receiving access to the sensitive to organization information.



Fig. 10. Reachability analysis using treemap:
(a) the initial position of the attacker is set to mail server; (b) the initial position of the attacker is set to firewall, the mail server vulnerability is mitigated.

In order to mitigate the given problem it is possible to update software and use Microsoft Exchange Server 8 SP1. In this case we could see that the security level of the mail server became green, what corresponds to secured state of the host.

To assess the usability of the designed GUI we used the requirements of the ISO/IEC 9126 standard purposed to evaluate the software quality [14]. When evaluating the GUI usability we mainly focused on metrics used to assess GUI attractiveness as the goal of the developed tool is to enhance the operator efficiency through effective graphical presentation of the data and possibility to investigate them interactively. And it was important for us to receive a feedback about chosen visual models, implemented interaction techniques, color scheme, etc. Additionally, we also included

functional metrics describing functionality of the application such as a possibility to create, modify network configuration, detect critical hosts and implement experiments "what-if". Table 2 shows an example of metrics included in questionnaire.

In order to implement GUI usability assessment we invited specialists in information security as well as experts in ergonomics and GUI design. It is worth noticing that we questioned both practitioners and scientific researchers.

First they were given a short introductory course on the VizSecAnalyzer. Afterwards they were given a set of simple tasks such as creation of network model for a given network description or modification of existing configuration, detection of the critical nodes and a questionnaire to get acquainted with the tool and then the experts had to rank the metrics from 0 to 5, where the higher value means better implementation. Additionally they had to assess their own competence in the subject. Later the questionnaires were processed in order to elaborate averaged rank for each metrics.

Table 2. Metrics used in questionnaire

Name	Description
Visual metrics	
Information Architecture and Hierarchy	addresses current screen layout and assesses conceptual structure of the information and its logical layout
Ease of Navigation	superscribes issues of navigation affecting data flow and user orientation, it is used to assess how user can navigate in the system in order to get necessary information
Iconography	evaluates current icon use per available user functions and data presentation (i.e. displaying types of network hosts, attack actions)
Color scheme	outlines current color scheme of text and graphic elements within the user interface to ease navigation, create emphasis, and warn users through the application
Visual models	assesses the use of visual models and interaction techniques associated with them for a given task
Customization of GUI	considers a proportion of the customizable elements in GUI
Functional metrics	
Configuration of the network and its hosts	assesses a possibility to configure network and software properties
Downloading and saving of network configuration from/to file and database	evaluates a possibility to download existing network configuration form database (or file) and save changes in current configurations
Implementation of the experiments "what-if"	assesses a possibility to implement experiments "what-if" by specifying different malefactor models, initial location or applying patches for vulnerable hosts

According to the overall ranking the quality of the tool is good (rank 4 of 5). Most of the experts marked a good choice of visual models. Apart from this some experts gave useful comments on further elaboration of GUI such as a possibility to manage object properties by setting up their visibility in Property Editor View.

We compared the VizSecAnalyzer with the tools described in Section 2 and implement similar functionality: Nv [9], GARNET [33], NAVIGATOR [3]. We left out approach to attack graph visualization suggested in [24] as there is no implementation for it. The rest described tools deal with policy assessment. Nv tool is designed to visualize vulnerability scanner reports thus it implements only a part of the VizSecAnalyser functionality. The GARNET and NAVIGATOR is probably the closest to our approach. These tools are aimed to analyze the attack reachability by presenting network domains using treemaps. The information detailing in these tools is limited to the domain level while in our tool the user can drill down to host level due to implemented interaction techniques. The reachability display of the Navigator is better than in our tool as it shows the attack deployment, but this lack is compensated by graph-based view of the VizSecAnalyzer and aggregation technique that allows aggregating network nodes to the domain level thus hiding excessive information. Though both tools provide a possibility to implement ”what-if” experiments, but our tool allows specifying the attacker model and provides more convenient interface for network configuring. Apart from this the VizSecAnalyser visualizes vulnerability reports highlighting the most critical assets. Among commercial tools the Network Vulnerability Manager of RedSeal company [28] exhibit the similar functionality. It also provides treemap-based view of vulnerability reports and allows identifying those vulnerabilities that can be accessed from threat sources to isolate exposure to attacks. However it does not allow implementing ”what-if” experiments taking the attacker model into account and does not display calculated security metrics based on attack graph analysis. The visualization of network traffic, e.g. [7, 12, 16-18, 27, 30], systems shows what is happening or has happened, but not the comparatively large set of what could happen.

5 Conclusion

In the paper we analyzed the visualization techniques used for network security assessment, suggested a common framework for SIEM visualization and developed the SIEM visualization component VisSecAnalyzer.

We described the proposed visual models and interaction techniques implemented in the VisSecAnalyzer, and presented case study for network level assessment and countermeasure plan adjustment. VisSecAnalyzer is intended for analyzing general security level, determining the most critical hosts and assessing consequences of exploitation of week places in security mechanisms. These data are used for prioritization of the needed countermeasures. The paper demonstrated the VisSecAnalyzer possibilities for the tasks of attack modeling and security assessment. Usability evaluation of the VisSecAnalyzer showed the quality of the tool is good.

The future research will be devoted to further elaboration and experimental analysis of suggested visualization techniques. We will evaluate the performance of proposed visualization system and assess the usability of the graphical user interfaces.

Acknowledgements. This research is being supported by grant of the Russian Foundation of Basic Research (project #13-01-00843-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.

6 References

1. Anwar, M., Fong, P.W.L., Yang, X.-D., Hamilton, H.: Visualizing Privacy Implications of Access Control Policies in Social Network Systems. Proc. of the 4th International Workshop on Data Privacy Management (DPM'09). LNCS, Vol.5939, pp.106-120 (2010)
2. Anwar, M., L.Fong, P.: A Visualisation Tool for Evaluating Access Control Policies in Facebook-style Social Network Systems. Proc. of the 27th Annual ACM Symposium on Applied Computing (SAC'2012). ACM, New York, USA, pp.1443-1450 (2012)
3. Chu, M., Ingols, K., Lippmann, R., Webster, S., Boyer, S.: Visualizing Attack Graphs, Reachability, and Trust Relationships with NAVIGATOR. Proc. of the Seventh International Symposium on Visualization for Cyber Security, Ontario, Canada, pp.22-33 (2010)
4. Common Vulnerabilities and Exposures. http://cve.mitre.org/
5. Common Vulnerability Scoring System. http://www.first.org/cvss/
6. Ficco, M., Romano, L.: A generic intrusion detection and diagnoser system based on complex event processing. Proc. of the 1st International Conference on Data Compression, Communication, and Processing. pp.275-284 (2011)
7. Fischer, F., Fuchs, J., Mansmann, F.: ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data. Proceedings of the Eurographics Conference on Visualization (EuroVis), pp.97-101 (2012)
8. O'Hare, S., Noe, S., Prole, K.: A Graph-theoretic Visualisation Approach to Network Risk Analysis. Proc. of the 5th International Workshop on Visualisation for Computer Security (VizSec'08), LNCS, Vol.5210, Springer-Verlag, Berlin, Heidelberg, pp.60-67 (2008)
9. Harrison, L., Spahn, R., Iannacone, M., Downing, E., Goodall, J.R.: NV: Nessus Vulnerability Visualisation for the Web. Proc. of the VizSec'12, October 15 2012, Seattle, WA, USA (2012)
10. Heitzmann, A., Palazzi, B., Papamanthou, C., Tamassia, R.: Effective Visualisation of File System Access-Control. Proc. of the 5th international workshop on Visualisation for Computer Security (VizSec'08), LNCS, Vol.5210, Springer-Verlag, Berlin, Heidelberg, pp.18-25 (2008)
11. Homer, J., Varikuti, A., Ou, X., McQueen, M.A.: Improving Attack Graph Visualisation through Data Reduction and Attack Grouping. Proc. of the 5th international workshop on Visualisation for Computer Security (VizSec'08), LNCS, Vol.5210, Springer-Verlag, Berlin, Heidelberg, pp.68-79 (2008)
12. Inoue D., Eto M., Suzuki K., Suzuki M., Nakao K.. DAEDALUS-VIZ: Novel Real-time 3D Visualization for Darknet Monitoring-based Alert System". Proc. VizSec '12, October 15, Seattle, WA, USA (2012)
13. Keim, D., Andrienko, G., Fekete, J.-D., Goerg, C., Kohlhammer, J., Melancon, G.: Visual Analytics: Definition, Process, and Challenges. Information Visualisation, LNCS 4950, Springer-Verlag, Berlin Heidelberg, pp.154-175 (2008)
14. Komiyama, T.: Usability Evaluation Based on International Standards for Software Quality Evaluation. Nec Technical Journal. Vol.3. No.2 (2008)
15. Kotenko, I., Chechulin, A.: Attack Modeling and Security Evaluation in SIEM Systems. International Transactions on Systems Science and Applications, Vol.8, December 2012, pp.129-147 (2012)

16. Lakkaraju, K., Yurcik, W., Lee, A. J.: NVisionIP: Netflow visualizations of system state for security situational awareness. Proc. of the ACM workshop on visualization and data mining for computer security (VizSEC/DMSEC ’04). New York, USA, pp.65-72 (2004)
17. Lau, S.: The spinning cube of potential doom. Communications of the ACM, Vol. 47(6), pp.24-26 (2004)
18. Lee, C.P., Trost, J., Gibbs, N., Beyah, N., Copeland, J.A.: Visual Firewall: Real-time Network Security Monitor. Proc. of the IEEE Workshop on Visualization for Computer Security (VizSEC 05), pp.129-136 (2005).
19. Mansmann, F., Göbel, T., Cheswick, W.: Visual Analysis of Complex Firewall Configurations. Proc. of VizSec'12, October 15, 2012, Seattle, WA, USA (2012)
20. Marty, R.: Applied Security Visualisation. NY, Addison Wesley Professional (2008)
21. Montemayor, J., Freeman, A., Gersh, J., Llanso, T., Patrone, D.: Information Visualisation for Rule-based Resource Access Control. Proc. of International Symposium on Usable Privacy and Security (SOUPS) (2006)
22. National Vulnerability Database. http://nvd.nist.gov/
23. Nessus vulnerability scanner website. http://www.tenable.com/
24. Noel, S., Jacobs, M., Kalapa, P., Jajodia, S.: Multiple Coordinated Views for Network Attack Graphs. Proc. of the IEEE Workshops on Visualisation for Computer Security, IEEE Computer Society, pp.12 (2005)
25. Noel, S., Jajodia, S.: Understanding Complex Network Attack Graphs through Clustered Adjacency Matrices. Proc. of the 21st Annual Computer Security Applications Conference (ACSAC’05). IEEE Computer Society, pp.160-169 (2005)
26. Novikova, E., Kotenko, I.: Analytical Visualization Techniques for Security Information and Event Management. Proc. of the 21th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2013). Belfast, Northern Ireland. Los Alamitos, California. IEEE Computer Society, pp.519-525 (2013)
27. Ohno, K., Koike, H., Koizumi, K.: IP Matrix: an effective visualization framework for cyber threat monitoring. Proc. of the 9th International Conference on Information Visualization (IV05), Washington, DC. IEEE Computer Society, pp.678–685 (2005)
28. RedSeal Networks Vulnerability & Risk Management Solution. http://www.redsealnetworks.com/solutions/vulnerability/
29. Stasko, J., Catrambone, R., Guzdial, M., McDonald, K.: An Evaluation of Space-filling Information Visualisations for Depicting Hierarchical Structures. International Journal of Human-Computer Studies, 53(5), pp.663-694 (2000)
30. Taylor, T., Brooks, S., Mchugh, J., Brooks, S.: NetBytes Viewer: An Entity-based Netflow Visualization Utility for Identifying Intrusive Behavior. In VizSEC 2007: Proc. of the 2007 Workshop on Visualization for Computer Security, pp.101-114 (2008)
31. Tran, T., Al-Shaer, E., Boutaba, R.: PolicyVis: Firewall Security Policy Visualisation and Inspection. Proc. of the 21st Conference on Large Installation System Administration Conference (LISA’07), USENIX Association, Berkeley, CA, USA, pp.1-16 (2007)
32. Williams, L., Lippmann, R., Ingols, K.: An Interactive Attack Graph Cascade and Reachability Display. Proc. of the Workshop on Visualisation for Computer Security, Sacramento, California, USA, Springer, Heidelberg, pp.221-236 (2007)
33. Williams, L., Lippmann, R., Ingols, K.: GARNET: A Graphical Attack Graph and Reachability Network Evaluation Tool. Proc. of the 5th International Workshop on Visualisation for Computer Security (VizSec'08), LNCS, Vol.5210, Springer-Verlag, Berlin, Heidelberg, pp.44-59 (2008)