Methodological Primitives for Phased Construction of Data Visualization Models

Maxim Kolomeec, Andrey Chechulin*, and Igor Kotenko
Laboratory of Computer Security Problems
St. Petersburg Institute for Informatics and Automation (SPIIRAS)
39, 14 Liniya, St. Petersburg, Russia
{kolomeec, chechulin, ivkote}@comsec.spb.ru

Abstract

The paper considers common methodological primitives for phased construction of data visualization models, which will help to create new graphical models of data security visualization, or will help to show advantages and disadvantages of existing models. The paper also considers examples of graphical models and additional tools, which allow to work with these models. The purpose of the paper is to form a comprehensive vision to create data security visualization models. The primitives classification and communication between them are suggested. On the base of identified primitives, graphical models and additional tools to work with graphical models, a methodology for constructing data security visualization models is provided. This methodology can be used for improving efficiency of existing models and for evaluating their effectiveness. The paper also considers a new visualization model for network security which was developed based on the proposed visualization process.

Keywords: data visualization models, security data visualization, network visualization.

1 Introduction

The construction of information security visualization models is a complex process, which is defined by different aspects which may belong to different knowledge domains such as computer graphics, mathematics, statistics, cognitive psychology and design [33, 2]. In the conditions of continuous increase of the volume and dimensionality of the visualized data [15], the problem of formation of new visualization models and techniques is very urgent. In order to develop new visualization models and techniques, it is necessary to know general features of their construction process and also be able to navigate in the existing methods, including those used outside the sphere of information security, where the authors carried out intensive research and develop the visualization subsystem.

Often new models of visualization systems come from the fields of art, media design, marketing, bioinformatics, etc. Despite the difference of scope, the concept of data display and representation of complex relations between them remain the same [7, 20, 3] (Figure 1): first of all data is collected, its analysis and filtering is performed, and only then data is converted into graphical primitives which determine the geometric data (the graphical model), that, together with the management tools and the display, represents the image data (the visualization model).

Papers dealing with methods of data visualization, as a rule, pay insufficient attention to the integrity of the process, namely, they do not consider important visualization aspects, do not mention what tools and libraries are used to implement the graphical model or what conceptual tools can enhance possibilities of the graphical model, and do not consider new and unique graphical models.

Journal of Internet Services and Information Security (JISIS), volume: 5, number: 4 (November 2015), pp. 60-84
*Corresponding author: Tel: +7-812-328-7181, Web: http://comsec.spb.ru/chechulin/

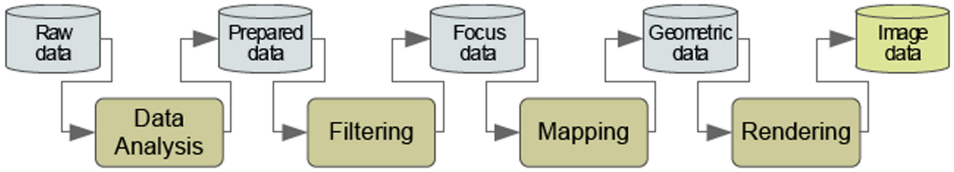


Figure 1: Visualization pipeline

For example, in [7] great attention is paid to the stage of data preparation, but visualization models themselves are considered in terms of ready-made tools. The stage of data preparation is certainly important, but exactly choice of the visualization model is the key for efficient interaction between data and users. [15] presents an overview of graphical models, however, main aspects on which one must make a choice of the model or develop one's own are not mentioned. [1] also presents a review of the traditional methods of visualization, but there are no new and unique concepts, construction of which is currently relevant. Separately, it is possible to select reviews of the areas not related to information security. For example, graphical models in sociologic domain [11] can be successfully transferred to the area of information security, but before that they need to be converted in accordance with the characteristics of data related to information security.

When building or choosing a model it is important to understand how the various stages and elements of the visualization process affect the model comprehensively. This overview describes some of the methodological primitives on the example of the stepwise model building for rendering prepared data with the aim to create a comprehensive vision of the model creation process and aspects influencing it. Considered primitives can be classified as:

- visualization process primitives – the aspects that affect the process of the visualization model building and with the use of which the initial metamodel is developed;
- primitives of graphical models – the basic principles to develop the model visualization; often it is the selection of the graphic concept defines the format of the interaction of data and users and it defines the limits and possibilities of extension of this interaction;
- additional tools – the components of tools that extend the capabilities of graphical models.

The paper is organized as follows: section 2 describes the primitives of the process of visualization, needed to be considered before design starts; section 3 outlines the primitives of graphical models; in section 4 we describe additional tools; section 5 discusses visualization libraries; section 6 shows an example of how by means of the primitives we developed a new visualization model; section 7 is the conclusion.

2 Visualization Process Primitives

Before designing your own or choosing a ready visualization model, it is important to understand the aspects within which this model needs to function. These aspects can be used as the elements of the template when building, and as the tools of choice for analysis of existing systems. They can be classified into two classes:

- aspects of information content – elements, in accordance with which it is necessary to build or choose a visualization model; aspects of information content set the requirements to the visualization model, affecting the information content;

- the aspects of efficiency – elements, in accordance with which it is desirable to construct the visualization model; efficiency aspects define the requirements to the model, affecting the efficiency of presenting information to the user.

The principles themselves and their sets may vary depending on the render target and resources available. If it is assumed that the visualization techniques will be used by users who do not have the required expertise, most likely, one should pay attention to aspects related to the design and, in general, to the aspects of effective presentation of information. In the case of information security, where the accuracy and completeness of information have a decisive role, one should pay attention to aspects of information content.

2.1 Examples of Aspects of Information Content

Aspects of information content effects the visualization model, setting limits on the amount of information that the model can display. Let us consider a few examples of the aspects that affect the information content of the model.

2.1.1 Visual Search Pattern

This principle was suggested by Ben Shneiderman in 1996 [25]. According to it, the visual search consists of three steps: (1) an overview of the situation as a whole; (2) scaling and filtering; (3) details on demand. Accordingly, any visualization model, which aims not only performance, but also the search and analysis of information, must have tools for each stage.

Figure 2 show an example of execution of this principle, based on the analysis of intercepted TCP packets.

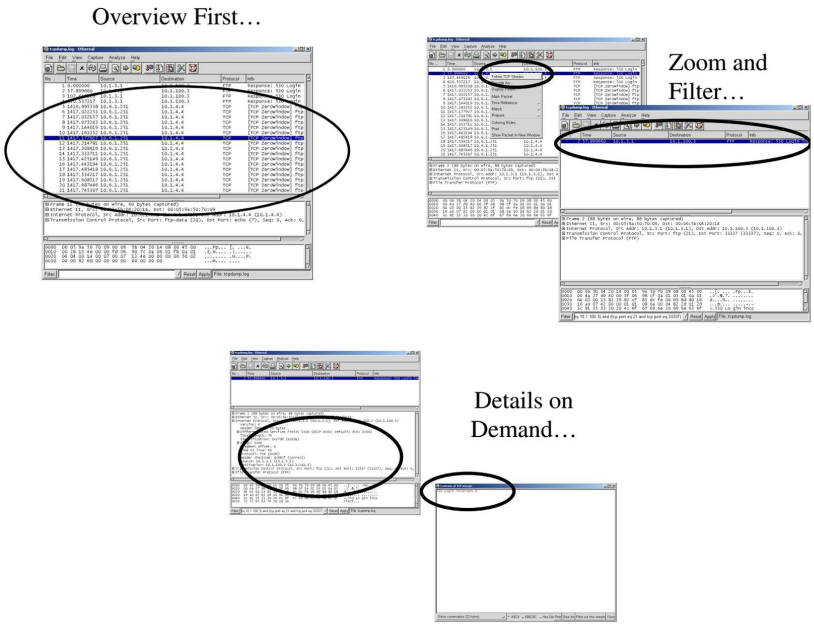


Figure 2: Visual search example

Initially the search is performed on the top level of abstraction, allowing to see the situation for all intercepted packets as a whole; in the second stage, packets are filtered so that the user can see the packets

in the framework of a specific connection; in the third stage, the user, on demand, can go to the lower level of abstraction to look at the details of the packet.

2.1.2 Cognitive Features

The efficiency of the visualization model is entirely defined by the cognitive features of the user (memory, perception, association, etc.). The visualization model must incorporate these features in general and not to go beyond the limits of their capabilities [8]. Despite the fact that this aspect belongs entirely to the domain of psychology, its foundation is often understandable on the intuitive level. But the larger and more complex graphical model is, the more influence of this aspect is. It is particularly important to take into consideration the cognitive features in the development of models that work with large or heterogeneous data. Examples are as follows:

- taking into account the psychological associations, red color is a “reserved color” for mapping threats (left part of Figure 3);
- the analysis of individual elements and metrics can be difficult, as the model goes beyond the capabilities of the cognitive features (the limits of perception of many elements)(right part of Figure 3).

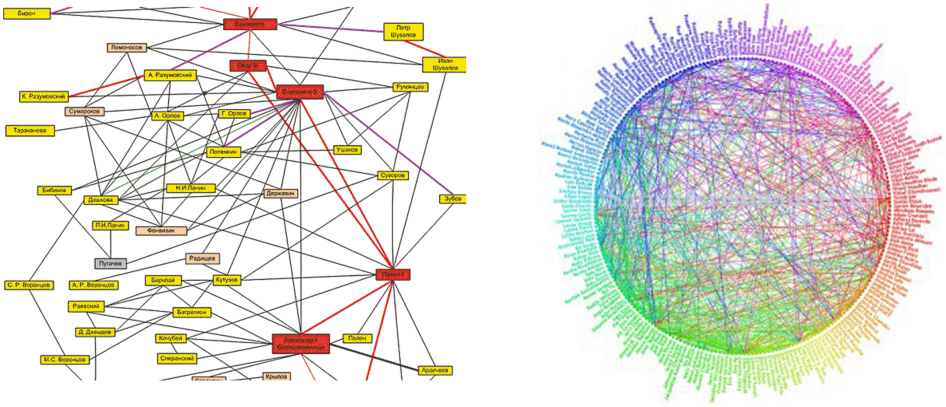


Figure 3: Graph with ”reserved color” and complex radial graph

2.1.3 Correspondence of Data and Visual Representation (Lie Factor)

In data analysis it is important that data visualization corresponds to the source. In some models (for example, using a three-dimensional space) proportional ratio between visualization and data could be broken.

In [30] Edward Tufte proposes to assess the degree of conformity of the data and the information depicted with the help of the coefficient “lie factor”, which is attitude of ”size of effect of visualization” and ”size of effect in data”.

Let us consider the example: in the infographics (Figure 4) [30], which aims to show how the fuel consumption per person in the USA increased, the value for 1978 is 18 gallons, and for 1985 is 27.5 gallons, i.e., the size of effect of data equal 53%; the road width, which is the visualization of the fuel consumption for 1978 is equal to 1 centimeter, for 1985 it equals to 7.8 centimeters; the size of effect of visualization is 780%; such a discrepancy between the performance efficiency and data efficiency gives the lie factor = 17.8 [30].

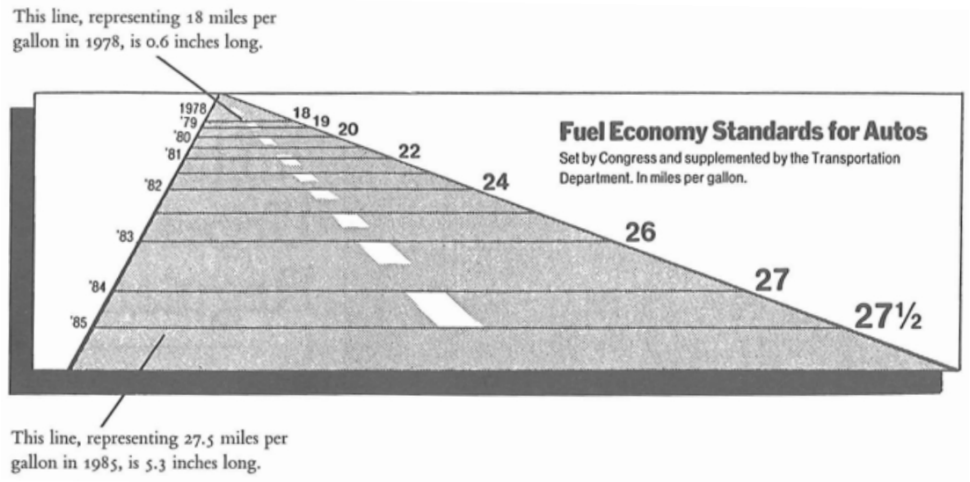


Figure 4: Infographics of fuel consumption per person in the USA

2.2 Examples of Efficiency Aspects

Aspects of efficiency have a greater influence on the graphical model than on the visualization model. They determine the efficiency of dialogue between the visualization mechanisms and the user within the visualization model. We consider several aspects affecting the efficiency.

2.2.1 Chart Junk

The optimal model is not only the one into which we have nothing to add, but from which nothing can be left to take away [31]. If the item is not included in the minimum set of visualization elements necessary to solve the task, it can be a source of noise [31]. Typically, these elements are elements of design that were not developed in accordance with the features of the visualization model.

- In the left part of Figure 5 the model becomes more difficult to interpret, due to the need to analyze data in three dimensions. In this case, the information noise is the dimension that shows depth.
- In the right part of Figure 5 the user can take the gradient inside the provinces as a model parameter (e.g., population density/temperature, etc.).

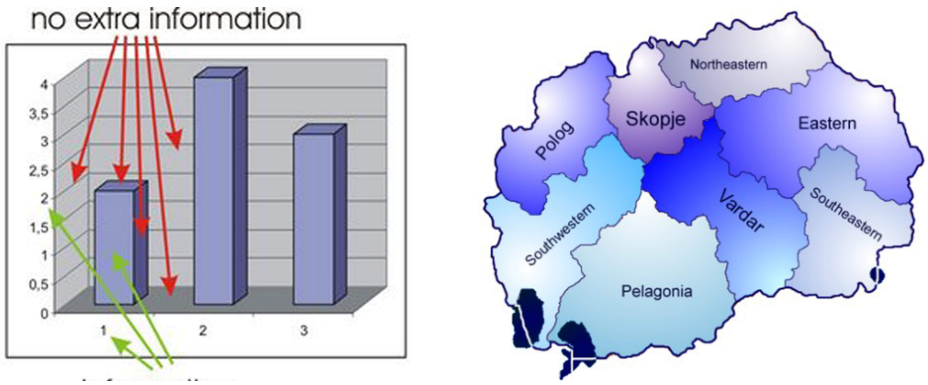


Figure 5: Models with Chart Junk

2.2.2 Direct Manipulations

The tools for interaction between information and users need to have counterparts in reality [12]. This approach uses human cognitive features and allows the user to predict what will happen after the interaction, to choose the tools and work with them intuitively.

For example, in the concept of tactile surfaces ”Material Design” [9, 23] each container is similar to the sheet of paper; the selected container through the shade is closer to the observer compared to the rest, making information about its activity clear at the intuitive level (Figure 6).

Figure 6: Tactile surfaces ”Material Design”

2.2.3 Graphic Design

The visualization model needs to work within one of the models of graphic design or have your own. The model of graphic design can also be represented as elements, each of which must have a rationale for its presence in the model. Often it is graphic design that has the greatest impact on the efficiency of interaction of the visualization model and the user. An example of the graph within the model of graphic design ”Material Design” [9, 23] is shown in Figure 7.

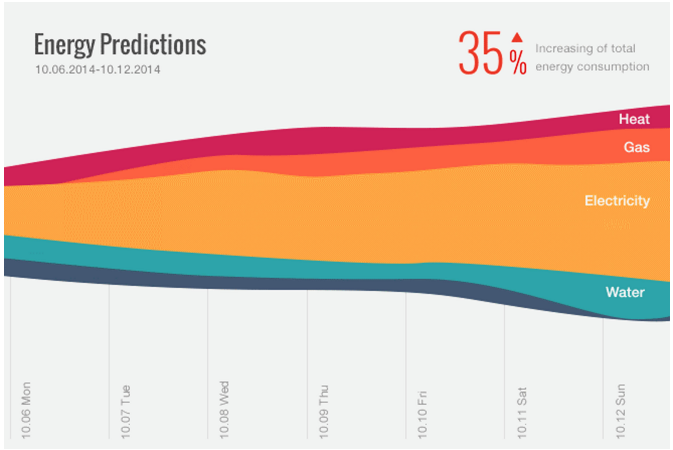


Figure 7: Graph within the model of graphic design

3 Primitives of Graphical Models

The visualization model is the core of the visualization process, determining most of its constraints and opportunities. As a rule, the choice of the model depends on the purpose of the visualization process and the type of information. On extent of use, graphical models can be classified as follows:

- wide spread;
- medium spread;
- specific.

3.1 Examples of Wide Spread Models

Wide spread models usually include: charts, graphs, their variations, and maps. The versatility of these models is due to the possibility of rendering almost any type of data.

3.1.1 Charts

Charts have proliferated as the simplest model of small data visualization [19]. In this case, the degree of success of visualization often depends on the choice of graphics, which, in turn, depends on the data type [21, 5]. Figure 8 shows some examples of simple charts.

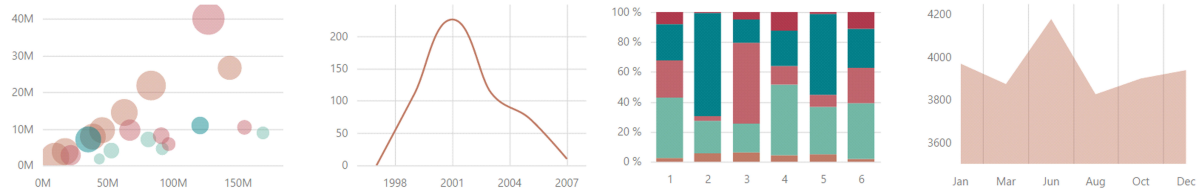


Figure 8: Examples of simple charts

3.1.2 Graphs

Graphs are especially popular in the information domain: they are intuitive, have many variations and are able to display large amounts of heterogeneous data [19]. Figure 9 depict some kinds of graphs:

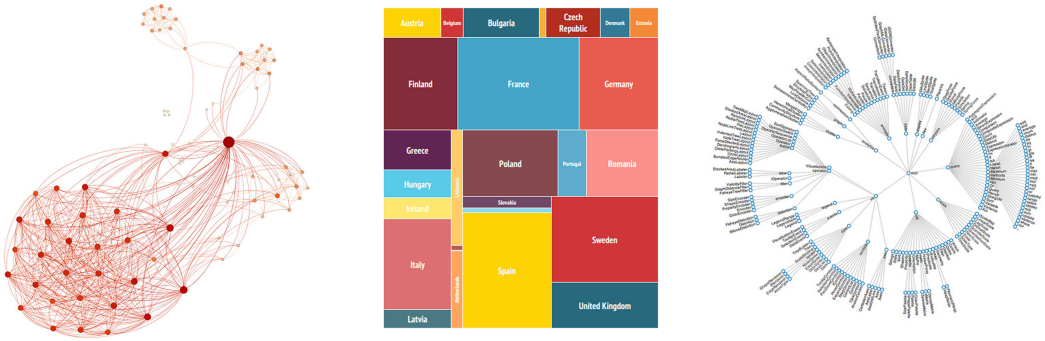


Figure 9: Examples of graphs

- classic standard regular graphs views (left part of Figure 9);
- maps of trees – hierarchical graphs on the plane, whose vertices are represented by rectangles and relationships by nesting (central part of Figure 9);
- radial hierarchical graphs, elements of which are arranged radially (right part of Figure 9).

3.1.3 Maps

Maps are used in cases where the geographic data are used as keys [19]. In this case, the other parameters are expressed in terms of other graphical models (charts, graphs, etc.), superimposed on model cards. Figure 10 shows a map of attacks [14, 17] in real time.

Figure 10: Geographical map of attacks

3.2 Examples of Medium Spread Models

Medium spread models are models developed in the framework of specific tasks, but have the potential in data visualization that goes beyond the purpose of their creation.

3.2.1 Matrices

In order to disguise an attack, the attacker usually modifies parameters identifying it, such as IP address. Thus, to identify the attacker, it is necessary to rely on other parameters, e.g. the arrival time of the packet, which depends on the type of operating system, router delay, and other metrics that are difficult to change.

In [18], to solve this task a sequential analysis based on the matrix representation is suggested.

In Figure 11 (left part) on the left in the matrix, where time is represented by a vertical scale, one of the bursts appear for several hours. The matrix on the center shows the activity of all the ports in the selected time range. Position of the splash on the matrix on the left corresponds to one abnormally active port in the Central matrix. The graphs on the right show that a small number of sources corresponds to a large number of recipients that, most likely, is a symptom of scanning the network.

In [18] it is also proposed to use this model in conjunction with classical random graphs for the analysis of network clusters (right part of Figure 11).

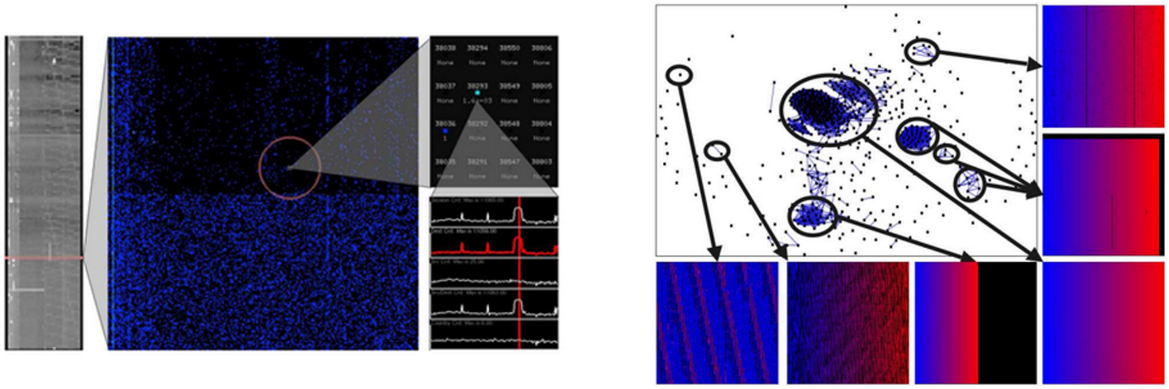


Figure 11: Models based on matrix representation (left) and usage of matrices for cluster analysis (right)

In [27] the distribution of TCP/IP addresses is analyzed for the purpose of reducing the risks of TCP/IP spoofing. Despite the fact that operating systems use random number generators for distribution, at their visualization, despite the high or complete randomness, for each operating system we can distinguish clusters of IP addresses, presence of which may help the attacker.

Figure 12 shows the analysis of the sequence of addresses generated by the operating system UNICOS 10.0.0.8. Despite the low assessment of the feasibility of the attack on the part of the operating system (right part of Figure 12), it is possible to allocate 3 big clusters within which the attack will probably be more successful.

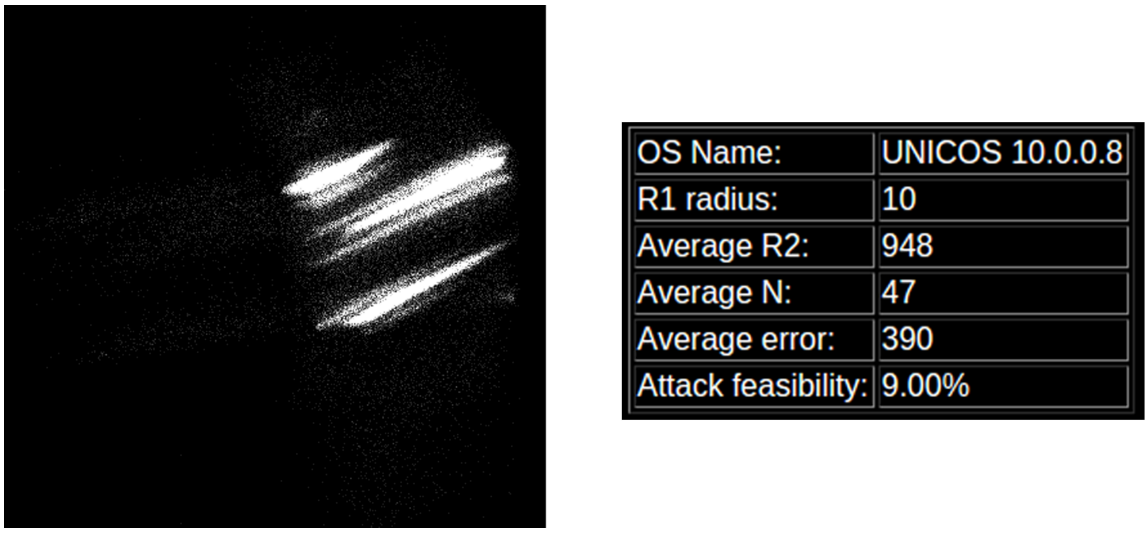


Figure 12: Matrix of TCP/IP addresses distribution (left) and Probabilistic estimation of the attack feasibility (right)

3.2.2 Histograms

When one wants to make general conclusions, matrix representation can be converted to histograms. The matrix in Figure 13 is obtained by using the parameter calculated on the basis of the analysis of visits using a unique email address, where red color corresponds to the maximum number, and black - to the lack of visits [18]. In histograms of two left matrices, similar outbursts are compared on the differences.

In histograms of the right two matrices, different spikes are compared for matches.

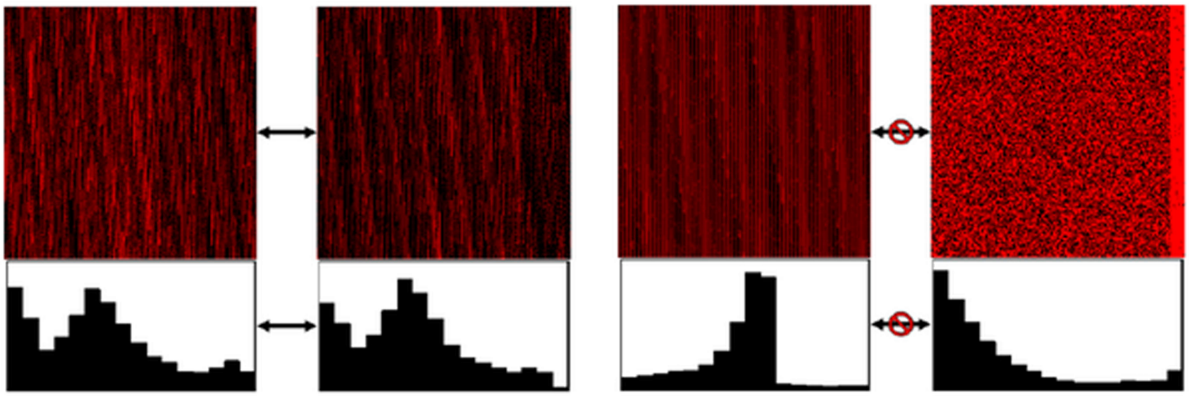


Figure 13: Representation of matrices as histograms

3.2.3 Trilinear Coordinates

A visualization model, proposed in [33], was developed basing on the triangular visualization model; it was presented by the US Geological Department. The model operates three different metrics, each of which corresponds to the edge of the triangle and is expressed in percentage in relation to each other. As an example there was taken the percentage of messages sent in different protocols (TCP, UDP, ICMP). The left part of Figure 14 depicts possible positions of the source in trilinear coordinates and the position of the source with parameters 30%, 40% and 30%. To display changes over time, for each migration of the source track a route is laid (right part of Figure 14). Areas in which the source spends most of the time, highlighted in blue. The model allows to identify anomalous behavior of the source when it enters the area that is not typical for its location or when the movement for it by itself is abnormal.

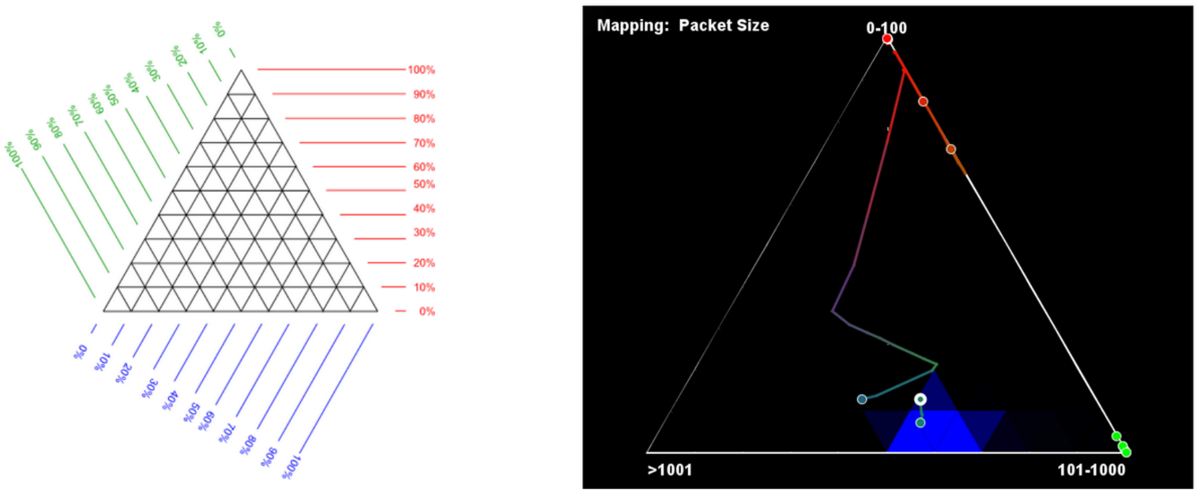


Figure 14: Models based on trilinear coordinates

3.2.4 Parallel Coordinates

The model of parallel coordinates [28] is a special case of graphics. It allows to efficiently display multidimensional data, with each data type along one of the parallels (Figure 15).

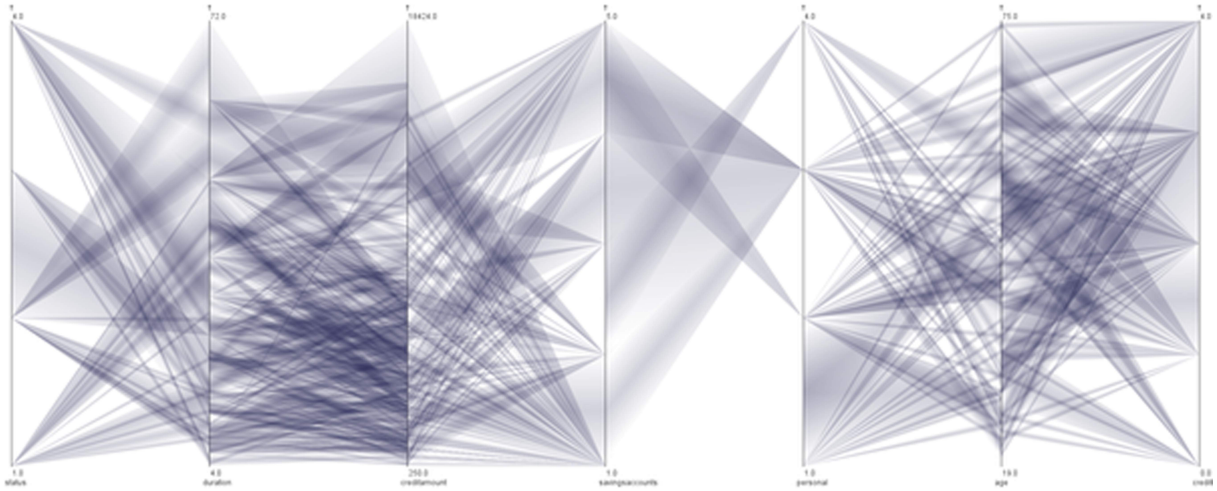


Figure 15: Model based on parallel coordinates

In [6] there was proposed a similar model, its axes are not parallel, but radial (Figure 16). It is assumed that the use of polar coordinates will allow to better identify abnormalities at the cognitive level, as well as to display metrics with greater efficiency.

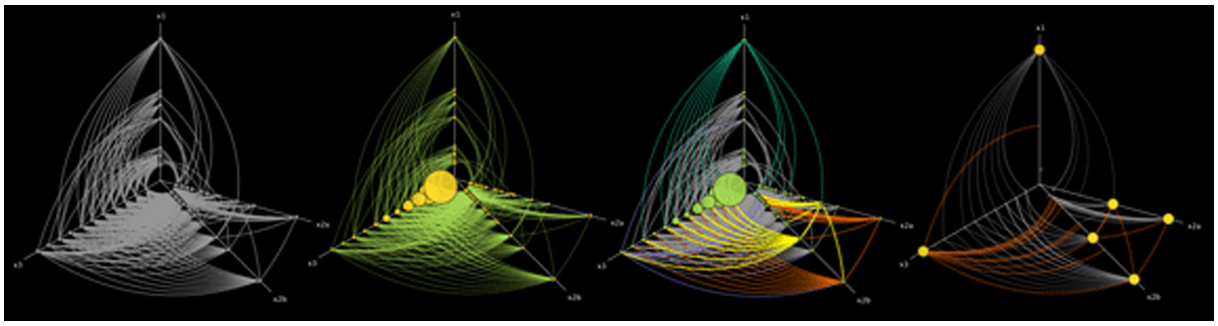


Figure 16: Model of radial coordinates

3.3 Examples of Specific Models

Specific models are models of a specialized type, designed for particular tasks, and which generally cannot be used for solving problems of another kind.

3.3.1 Visualization of Tools for Work with Log Files

Paper [26] offers visualization tools for detailed work with the log files based on regular expressions (Figure 17). The advantages of this approach are: the ability to work with abstractions (visualization elements) and with the log files simultaneously, quick navigation without the loss of orientation, analysis of subsets of records using the context and the research nature of the search instead of standard search queries.

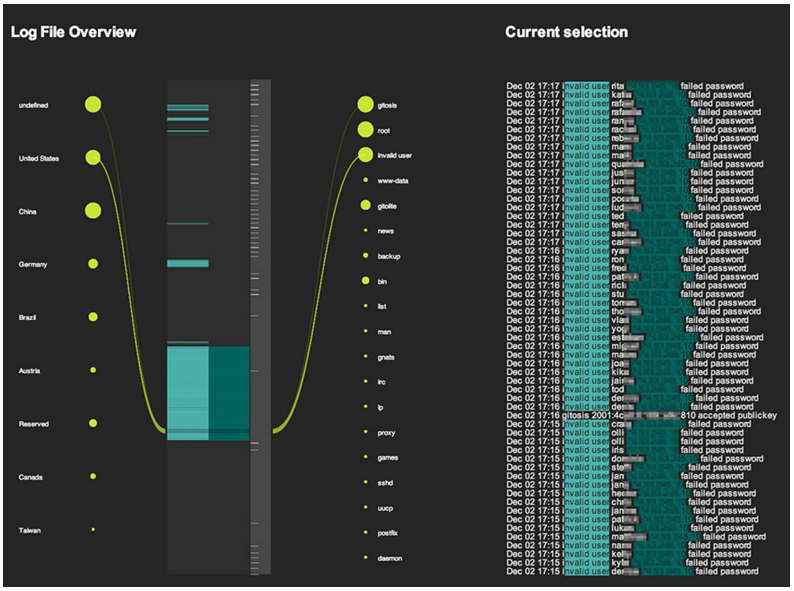


Figure 17: Visualization tools for work with the log files

3.3.2 Visualization of Complex Attacks

Paper [35] proposes a three-dimensional model to visualize complex multi-stage attacks (Figure 18). Each cylinder represents an event whose type is set to color: green – for scanning, purple – for remote access attempt, red – for successful attempt of remote access, yellow – for DoS. The severity level of alerts is set by the height of the cylinder (top left part of Figure 18).

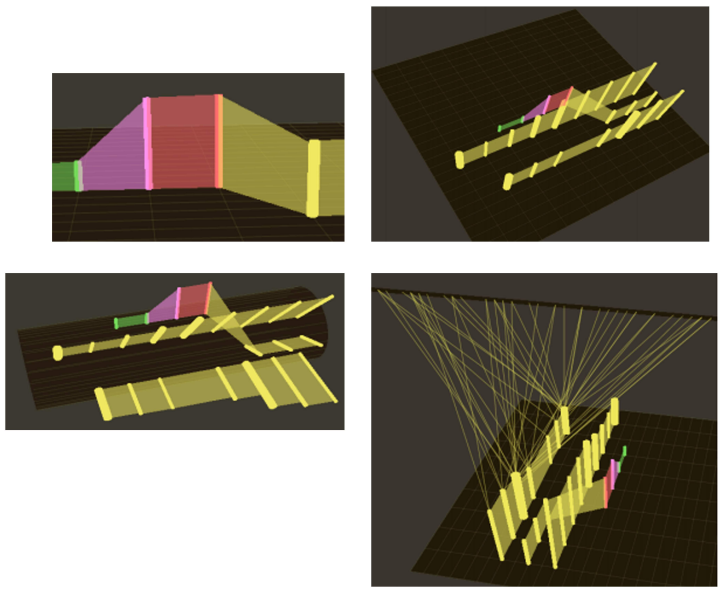


Figure 18: Visualization of complex attack

The key feature of the system is the possibility of a gradual tracking of the attack as a whole, knowing what events preceded it and what came after (top right part of Figure 18).

The system also has a number of additional tools, such as the transition from Cartesian to polar coordinates (bottom left part of Figure 18), possibility of showing the source of attacks in accordance with the type of the attack (bottom right part of Figure 18), and the management plane of the sources for best viewing angle.

4 Auxiliary Tools

For some data types or a particular set of data, the choice of a graphical model can be represented with just a few kinds of models or no models at all. For example, to represent a network topology, an obvious option is always to view the graph. However, if the multidimensionality of the data or their options are characterized by high volume, the visualization model at some point may go beyond the restrictive limits of the visualization process. Most sharply this problem appears in the visualization of large networks, when the user is forced already at the stage of the review to sacrifice with displaying of some of the metrics to dipict the others. In the left part of Figure 19, to display the individual elements, the user, using a zoom tool, has to refuse from displaying the network as a whole (right part of Figure 19). Thus, it is possible to select a class of primitive tools that cannot be independent graphical models, but can extend an existing model without violating its principles.

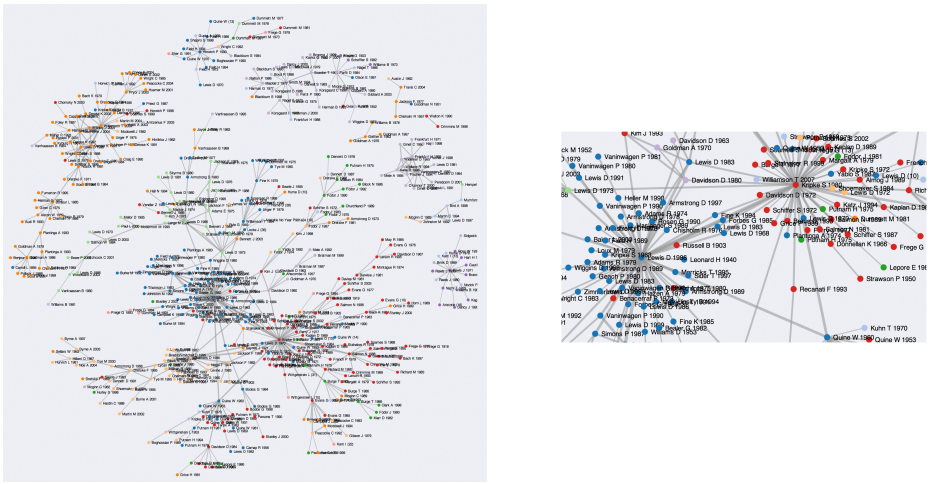


Figure 19: Zoomed graph segment (left) and overloaded graph (right)

4.1 Fisheye View

The main parameter affecting the perception efficiency when visualizing graphs is the number of nodes and connections between them. Classical solutions to problems with the number of nodes and connections is transition to a higher level of abstraction (which requires the development of the representation model of a higher level or relationships between the levels) or zooming, which increases one part of the graph with entire lost of another one, which disturbs vision of the situation in general. Sarkar and Brown [24] suggest using the effect of “fisheye” instead of scaling (Figure 20).

It is assumed that the user is able to select an area of focus, resulting in that interesting elements will become larger while others will remain in view. Unlike conventional scaling, the user does not have to

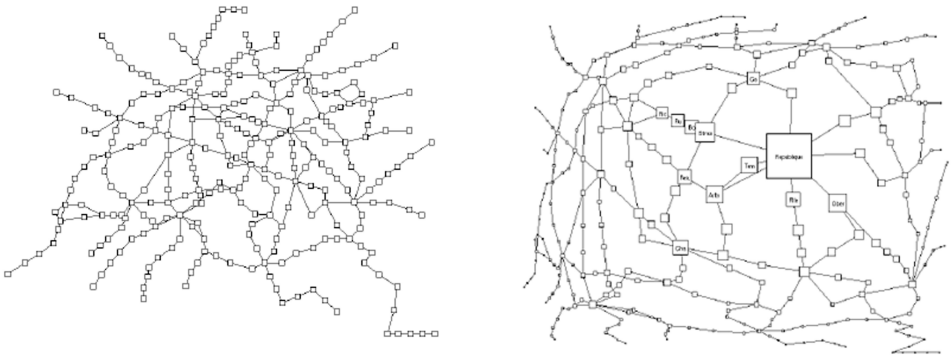


Figure 20: Graph without lens (left) and graph with lens (right)

mentally combine the images for a full overview. In the ”fisheye” tool the distortion factor corresponds to magnification of the scale.

4.2 Multiple Views

For some data types it is useful to use several visualization models simultaneously [32]. Finding necessary relations or anomalies in this approach is significantly simplified. In the case of multidimensionality of the metrics, the approach showing for each set of parameters the corresponding optimal graphical model may be more efficient than standard tools, such as the transition between levels of abstraction or exclusion of a number of parameters.

In Figure 21 the same data set is represented by parallel coordinates, 2D graph and trees map.

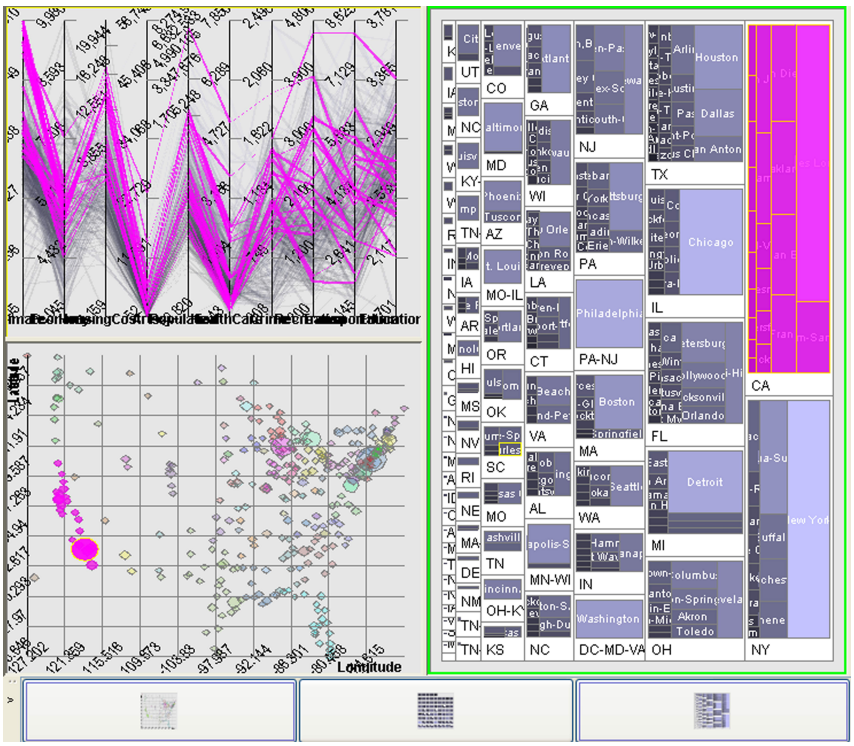


Figure 21: Different representations of the same graph

4.3 Tool ”Small Differences”

Sometimes, while applying filters, it is necessarily to see not only the result of filtration, but its preliminary results or multiple results at once. For example, in Figure 22 in the left part the set of results of filter applications with different parameters is displayed. Small components, with the size of a postage stamp, designed for instant answer to the question “what if?”, can not only speed up the process of finding information, but also to eliminate the need to uniquely choose the visualization model, where one of the metrics will be a preliminary result [34, 29].

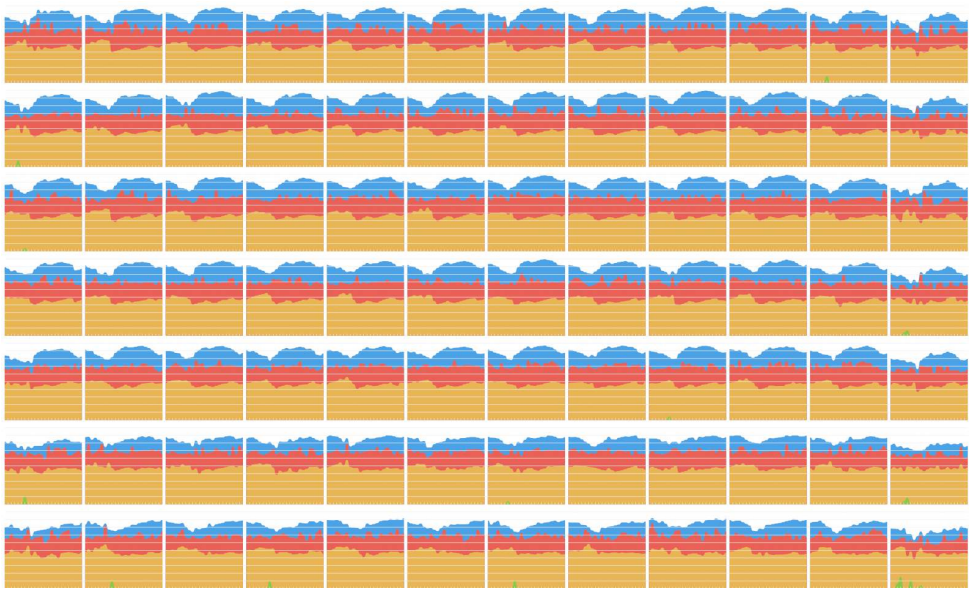


Figure 22: Representation of variations of filter application

5 Visualization Libraries

Constructing the visualization model is always dependent on the metrics, some of which are frequently encountered in many existing systems. Thus, it may turn out that the developer would implement something that was developed by others previously. Knowledge of basic data visualization libraries and patterns of implementation based on them is necessary before building the visualization models. The libraries also contain ready implementations of some components of the system and provide various frameworks to simplify the development process.

Below we consider some of popular non-profit libraries for the languages Java and JavaScript.

5.1 Jung

Simple and stable library Jung [13] is developed in Java and is mainly used to visualize graphs. Among the drawbacks of this library one can note that the developers did not adequately prepared documentation, and that the project stopped its development. So Jung can mostly be useful for visualizing simple graphs (Figure 23).

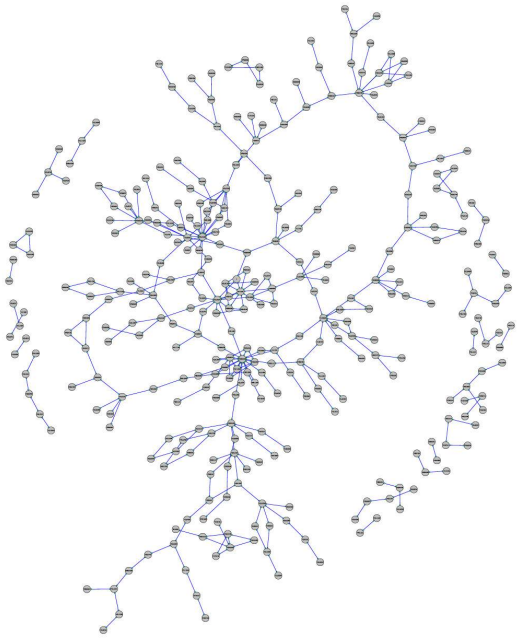


Figure 23: Example of a graph implemented in Jung

5.2 GraphViz

Visualization within the GraphViz software package [10] is performed using graph description language DOT [10]. GraphViz accepts a file in the DOT language, and automatically generates an image with a predetermined model. The advantages of this library include a well-crafted documentation, ease of use, availability, clustering and support for multiple graphical models. The shortcomings are that GraphViz can only be used for the visualization of graphs. An example visualization of a graph with a partition into clusters is shown in Figure 24.

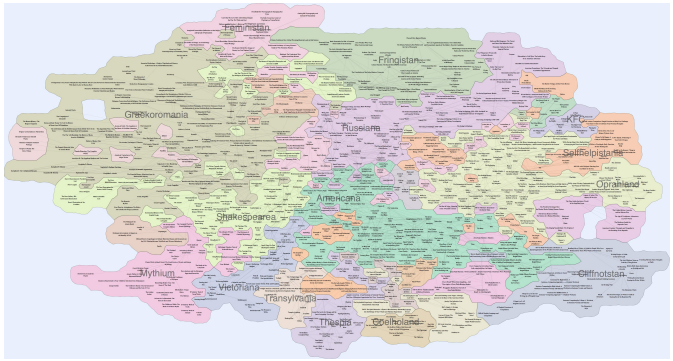


Figure 24: Example of a graph visualized with Graphviz

5.3 Prefuse

This library uses the standard Java library Java2D, so it is easy to integrate with applications developed using the Java Swing package. Prefuse [22] contains a wide range of tools (including a connection to the database, own data structures, support for animation, dynamic queries, search, tables, graphs, etc.) and, consequently, is suited not only for the visualization of graphs. Prefuse is a rather flexible tool, but its development is suspended, with the consequence that the documentation is not designed well enough. Example of a graph using Prefuse depicted in Figure 25.

Figure 25: Example of a graph implemented in Prefuse

5.4 D3

This library [4] was developed in JavaScript and is one of the most popular libraries for data visualization. It can be used as the basis to implement the visualization model within the framework of Data-Driven-Document, or use one of the hundreds of ready-made implementations of graphical models proposed by developers. This library has a well designed documentation, and this documentation is translated into many languages. Examples of graphical models implemented in D3 depicted in Figure 26.

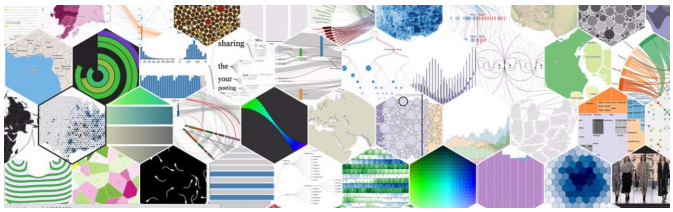


Figure 26: Examples of different models implemented in D3

6 Using Primitives to build a Visual Model based on Voronoi Diagram

Let us to build a graph-based visualization model based on methodological primitives of visualization process.

The process of the visualization model construction is depicted in Figure 27. It includes the following steps:

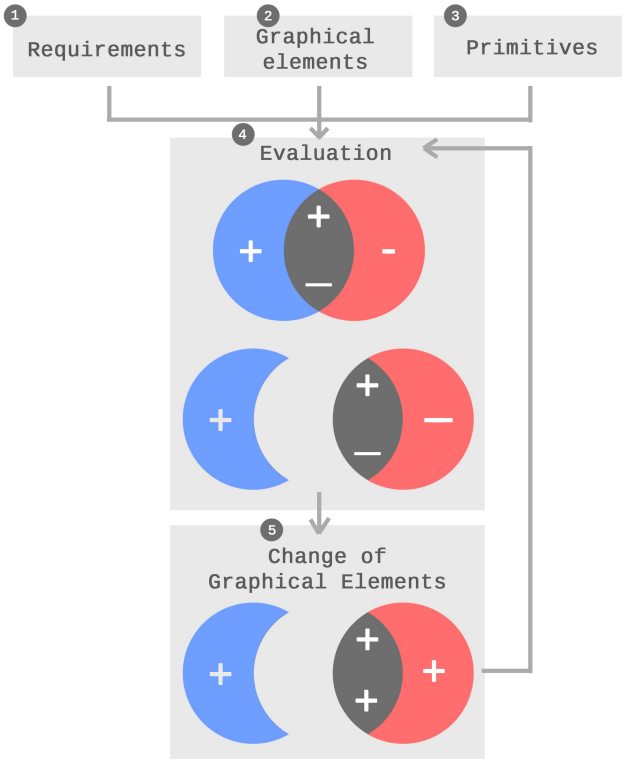


Figure 27: The process of visualization model construction

1. Forming the visualization target. This step is necessary to understand the purpose of the visualization model. For example, to visualize computer network, the following objectives are implemented: the ability to visualize the network topology; the ability of clustering; the ability to display more metrics compared with the existing graphical model (in our case - the graph). As a result, a list of requirements to the graphical model should be formed (block 1 in Figure 27). Here it is necessary to understand which requirements have higher priority, and which have a lower priority.

2. Identifying the graphical elements the graphical model contains (block 2 in Figure 27). For example: a table visualization consists of rows, columns and text; the graph consists of lines, dots, geometric figures, etc.; geographic visualization consists of 3D models of real physical objects, a map legend, geometric elements of the map, etc.

3. Forming the list of primitives, which were considered in the paper (block 3 in Figure 27).

4. As a result, the designer of the visualization model has a list of requirements to the graphical model, a list of graphical elements and a list of methodological primitives. At this step the designer

evaluates each element, using the methodological primitives. Each evaluation must represents conformity or nonconformity to achieve the purpose of the visualization model. It is one of the most important steps in the process of constructing the visualization model. Evaluation process is outlined in Figure 41 as block 4. The set on the top of the block contains 3 sections which are obtained as a result of the evaluation. Graphical elements that help in achieving the purpose of the visualization model are depicted as pluses. Graphical elements that prevent in achieving the purpose of the visualization model are depicted as minuses. Red and black sections contain minuses, so they are separated from the blue set (bottom part of block 4 in Figure 27) and will be replaced during the next step.

5. As a result of the previous step the designer has two crossed sets of graphical elements: the set of elements that helps in achieving the purpose of the visualization model and the set of elements that does not allow to achieve it. It is also possible to differentiate the intersection of these two sets, that has both of these characteristics depending on a particular primitive that has been used. In the set that does not allow to achieve the purpose the designer needs to change (replace) some graphical elements. For the intersection, such change should be at least considered, but not necessarily fulfilled, because the deficiencies can be offset by using auxiliary tools. The change should also be carried out using methodological primitives and auxiliary tools. As a result, a new set of graphical elements of the visualization model will be formed (block 5 in Figure 27).

6. A visualization model is analyzed as in step 4 using the same purpose and methodological primitives. If the model does not meet the goals, steps 5 and 6 should be repeated.

As an example of the considered process of the visualization model construction, let us develop a visualization model for representing a physical computer network.

Let us consider firstly the representation of a computer network as a graph (Figure 28).

Step 1. Setting a purpose we will use to evaluate the visualization effectiveness: it is needed to improve the network representation as a graph so that it could include a large number of security metrics and also have an ability to visualize the network topology.

Step 2. Forming the list of graphical elements of the graph, that can display security data: vertex color, vertex size, vertex opacity, vertex shape, rib thickness, color, edges, transparency, edges, clustering vertices by adding new elements.

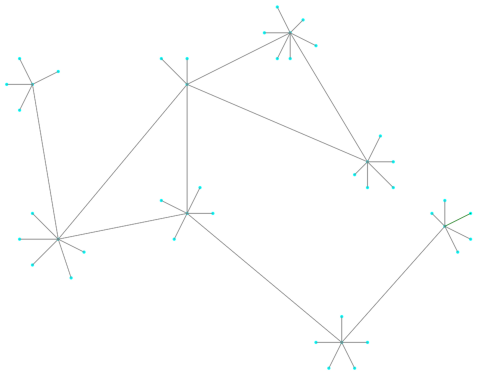


Figure 28: Visualization of a physical network using the graph

Step 3. Determining the list of primitives, including aspects of information content and efficiency aspects, auxiliary tools and graphical models. The whole list of used primitives is depicted in Table 1.

Type	Category	Title
Visualization process primitives	Aspects of Information Content	Visual Search Pattern, Cognitive Features, Lie Factor
	Efficiency Aspects	Chart Junk, Direct Manipulations, Graphic Design
Graphical Models	Wide Spread	Graphs, Maps, Charts
	Medium Spread	Matrices, Histograms, Trilinear Coordinates, Parallel Coordinates
	Specific Models	Tools to work with Log Files, Complex Attacks
Auxiliary Tools	-	Fisheye View, Multiple Views, Small Differences

Table 1: The list of primitives for the visualization model construction

Step 4. This step consists in evaluation of graphical elements of the graph. Visualization of the physical computer network using graphs contains a small number of graphical elements possible to display metrics (vertex color, vertex size, vertex opacity, vertex shape, rib thickness, color, edges, transparency, edges, clustering vertices by adding new elements). We consider them in detail from the point of view of the human cognitive features (section 2.1.2). For example, some elements of the graph cannot use all of the color ranges and all levels of transparency, as with the use of bright colors or transparency less than 30% the graphical elements will be poorly visible and, consequently, the edge or vertex of the graph may not be noticed by the user. Let us consider the graph from the point of view of the visual search pattern (section 2.1.1). Large graphs are difficult to scale and can display a small number of metrics, because graphs possess a small number of graphical elements. Let us consider the graph also from the point of view of information noise (section 2.2.1). When visualize the graph a large amount of unused space remains. This unused space can also be regarded as a kind of informational noise. Also, for clustering (the selection of groups of hosts) at the level of topology in graphs, as a rule, the developers introduce additional graphical elements (combine the nodes or set a different background color in the group) or operate former graphical elements (move the nodes to form grouping). Thus, when clustering the visual representation is changing dramatically, and the introduction of additional graphical elements makes the perception more difficult. As result, we define the following graphical elements to be replaced: graph edges, graph nodes, void in graph, vertex color, clusterization elements.

Step 5. On the basis of information about these shortcomings, we attempt to convert the graph so that the graph could show all the color ranges, the number of metrics increases, and entire available space is used. For this it is suggested to change the graphical relationship between nodes (graph edges) into the spatial relation (contacting nodes), and the nodes themselves to be increased in size so that they fill all available space. Thus, a special graphical model is developed, resembling a Voronoi diagram (Figure 29).

The Voronoi diagram includes the following graphical elements and visual possibilities: the color of the cell; cell size; cell transparency; the location of the cells relative to each other (shared side - evidence of a connection of computers); color of common edges of the cells; the total thickness of the edges of the cells; the transparency of the common edges of the cells; free space inside each cell that can be used for the representation of information; the shape of the cells; nesting of cells; separation of clusters of cells;

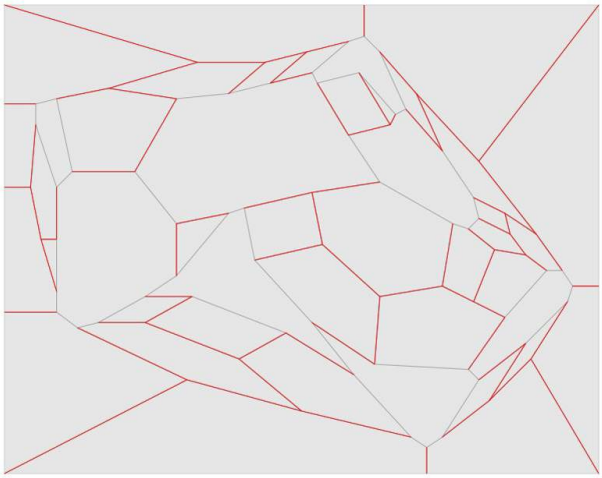


Figure 29: Visualization of a physical network based on the Voronoi diagram

moving the cells along the Z-axis (the extension of cells on different height relative to the screen plane in three-dimensional space).

Step 6. Now let us consider the new graphical model from the point of view of primitives of the visualization process. In the Voronoi diagram computers are represented in the form of cells of arbitrary shape lying in the plane. Communication between computers is represented by the relations of cells: if the cells are contacted (share at least one common edge), the computers, corresponding to the respective cells, are connected. Thus, it is formed one common cellular figure, which shows the topology of the computer network. At this the polymorphism of cells is provided preserving the topology: different parameters of computers can be set relative to each other by cell size, which ensures that the data and their representations correspond (section 2.1.3). In the case of increase or decrease of one of the cells, others will decrease or increase so that the relations (common edges) will not disappear. This will allow maximum use of space, and will thereby increase the efficiency of perception of the cognitive features (section 2.1.2). Also, in the case of nesting (when computer networks contain subnets), subnets can be displayed within a single cell, through which the entry to the subnet is provided. Thus the visual search pattern is implemented (section 2.1.1). This subnet will represent a Voronoi diagram of a smaller size that will ensure the integrity of the overall picture. As in the developed visualization there is no unused space and each element displays the needed information, and the control of noise is provided (section 2.2.1). To highlight items the idea of tactile surfaces through direct manipulation is used (section 2.2.2) within the graphic design ”Material Design” (section 2.2.3). As the Voronoi diagram is built directly from the graph, for final visualization one can use the tool ”multiple view” (paragraph 4.2), comparing the visualization in the form of a graph and the visualization of the Voronoi diagram. Thus, a new visualization model has no drawbacks that were identified during the analysis of graphs. Our model allows to make a new look at the network topology: the network can be represented in the form of a labyrinth with rooms-cell, thereby allowing to effectively visualize the number of steps to achieve a particular host (Figure 30). Inside each cell it is also possible to place information such as a subnet information, thereby showing nesting. The number of represented metrics was increased in comparison with the graph, and due to replacing the vertices in the plane new tools and methods for outlining parameters were appeared. The new visualization uses the cognitive features of human perception, thereby improving the efficiency of information representation.

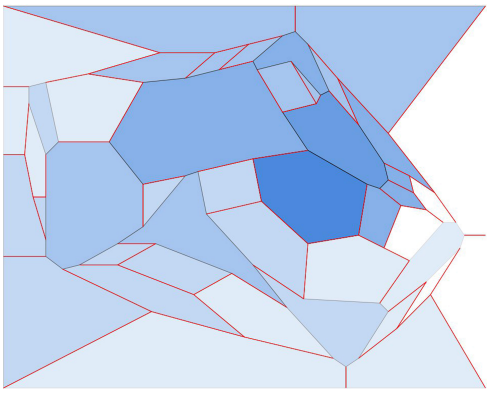


Figure 30: Visualization of the number of steps for achievement of hosts. The more steps are necessary to be done, the lighter the cell color is

7 Conclusion

We can say that the aim of the visualization process is to find a compromise between the data information content and efficiency of their representation. Each element in the visualization model, the aspects of the visualization process, the graphical model, the visualization tool, and the implementations library separately affect these indicators. But ultimately, the decision on the use of certain elements depends on the data. It is important to understand that it is not the data should be determined by the visualization model, but the visualization model should be generated on the basis of the data. Sometimes, the more appropriate set of elements is a simple and minimal set than a complicated graphical model with a large number of instruments under the control of a plurality of primitives.

In the paper we reviewed the main elements of the visualization process and showed the process of their application to develop the new visualization model. The resulting model has a significant number of advantages, satisfies the visualization primitives and can be used alongside with graphs. Possible tools and features of the representation of the graph by the Voronoi diagram still worth exploring that will be done in subsequent papers.

On the example of the resulting new visualization model, we can say that knowledge of the visualization primitives for the analysis of existing models and the ability to apply them at different stages of development of the visualization model will help if not to finally achieve the compromise of information content and efficiency, but to closely approach it.

Methods described in the paper have been successfully used by the authors both to visualize general models of attacks and the security metrics [16] and to visualize individual elements of the attacks presented in the CAPEC database [17].

Acknowledgement

This research is being supported by grant of the Russian Foundation of Basic Research (projects N 13-01-00843, 14-07-00697, 14-07-00417, 15-07-07451, 15-37-51126) and by the grant 15-11-30029 of the Russian Science Foundation in SPIIRAS.

References

[1] D. Barrera. Classifying and selecting appropriate security visualisation techniques. Master's thesis, Carleton University, Ottawa, Canada, September 2009.

[2] M. V. Botja. Infographic as object of information design. In *Proc. of the 8th Scientific-Practice Conference on New Information Technologies in Education, Ekaterinburg, Russia*, volume 8, pages 411–414. Russian State Vocational Pedagogical University, March 2015.

[3] F. D. and D. D. Security visualization survey. In *Proc. of the 12th Colloquium for Information Systems Security Education University of Texas (CISSE'08), Dallas, Texas, USA*, pages 119–126. CISSE, June 2008.

[4] Official web site of d3js library. `www.d3js.org`. [Online; Accessed on November 2, 2015].

[5] Devexpress company blog. `www.habrahabr.ru/company/devexpress`. [Online; Accessed on November 2, 2015].

[6] S. Engle and S. Whalen. Visualizing distributed memory computations with hive plots. In *Proc. of the 9th International Symposium on Visualization for Cyber Security (VizSec'12), Seattle, WA, USA*, pages 56–63. ACM, October 2012.

[7] L. Falschlunger, O. Lehner, C. Eisl, and H. Losbichler. Development of a data visualization model based on information processing theory. In *Proc. of the 9th conference for Austrian universities of applied sciences, Hagenberg, Austria*, pages 1–7, April 2015.

[8] B. Goldstein. *Cognitive Psychology*. Thomson Wadsworth, 2005.

[9] Google inc. material-design introduction. `www.google.com/design`. [Online; Accessed on November 2, 2015].

[10] Official web site of graphviz library. `www.graphviz.org`. [Online; Accessed on November 2, 2015].

[11] K. Healy and J. Moody. Data visualization in sociology. *Annual Review of Ecology and Systematics*, 40:105–128, June 2014.

[12] E. Hutchins, J. Hollan, and D. Norman. Direct manipulation interfaces. *Human-Compuner Interaction*, 1:311–338, 1985.

[13] Official web site of jung library. `www.jung.sourceforge.net`. [Online; Accessed on November 2, 2015].

[14] Kaspersky cyberthreat real-time map. `www.cybermap.kaspersky.com`. [Online; Accessed on November 2, 2015].

[15] J. S. Klyshinskij, S. V.Rysakov, and A. I. Shihov. Review of the methods of multidimensional data visualization. *New information technologies in automated systems*, 17:519–530, 2014.

[16] I. V. Kotenko and A. A. Chechulin. A cyber attack modeling and impact assessment framework. In *Proc. of the 5th International Conference on Cyber Conflict 2013 (CyCon'13), Tallinn, Estonia*, pages 119–142. IEEE, June 2013.

[17] I. V. Kotenko, E. V. Doynikova, and A. A. Chechulin. General enumeration and classification of attack patterns (capec): description and application examples. *Information Security, Inside*, 4:54–56, 2012.

[18] K.-L. Ma. Cyber security through visualization. In *Proc. of the 2006 Asia-Pacific Symposium on Information Visualisation (APVis'06), Tokyo, Japan*, volume 60, pages 3–7. APVIS, February 2006.

[19] R. Marty. *Applied Security Visualization*. Addison Wesley Professional, 2009.

[20] E. S. Novikova and I. V. Kotenko. Analytical visualization techniques for security information and event management. In *Proc. of the 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'13), Belfast, Nothern Ireland*, pages 519–525. IEEE, February 2013.

[21] E. S. Novikova and I. V. Kotenko. Visualization of security metrics for cyber situation awareness. In *Proc. of the 1st International Software Assurance Workshop (SAW'14), Fribourg, Switzerland*, pages 506–513. IEEE, September 2014.

[22] Official web site of prefuse library. `www.prefuse.org`. [Online; Accessed on November 2, 2015].

[23] Redmadrobot company blog. `www.habrahabr.ru/company/redmadrobot`. [Online; Accessed on November 2, 2015].

[24] M. Sarkar and M. H. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12):73–83, 1991.

[25] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc.*

of the 1996 IEEE Symposium on Visual Languages (VL'1996), Boulder, CO, USA, pages 336–343. IEEE, September 1996.

[26] J.-E. Stange, M. Dörk, J. Landstorfer, and R. Wettach. Visual filter: Graphical exploration of network security log files. In *Proc. of the 11th Workshop on Visualization for Cyber Security (VizSec'14), New York, NY, USA*, pages 41–48. ACM, November 2014.

[27] Tcp/ip sequence number analysis. `www.lcamtuf.coredump.cx/newtcp`. [Online; Accessed on November 2, 2015].

[28] S. Tricaud, K. Nance, and P. Saade. Visualizing network activity using parallel coordinates. In *Proc. of the 44th Hawaii International Conference on System Sciences (HICSS'11), The Grand Hyatt Kauai Resort and Spa Kauai, USA*, pages 1–8. IEEE, January 2011.

[29] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[30] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1991.

[31] E. Tufte. *Visual Explanations*. Graphics Press, 1997.

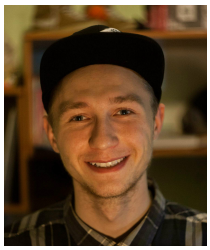
[32] M. Wang and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. of Conference on Advanced Visual Interfaces (AVI'00), Palermo, Italy*, pages 110–119. ACM, May 2000.

[33] R. B. Whitaker. Applying information visualization to computer security applications. Master's thesis, Utah State University, January 2010.

[34] L. Wroblewski. Small multiples within a user interface. *UXmatters*, December 2005.

[35] A. Yelizarov and D. Gamayunov. Visualization of complex attacks and state of attacked network. In *Proc. of the 6th International Workshop on Visualization for Cyber Security (VizSec'09), Atlantic City, NJ, USA*, pages 1–9. IEEE, October 2009.

Author Biography



Maxim Kolomeec is currently pursuing Specialist degree in Information Security at the Saint-Petersburg Electrotechnical University ”LETI”, Russia. He is working as a developer at the Laboratory of Computer Security Problems of St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). His research interests include distributed system security, and security visualization.

Andrey Chechulin received his B.S. and M.S. in Computer science and computer facilities from Saint-Petersburg State Polytechnical University and PhD from St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) in 2013. In 2015 he was awarded the medal of the Russian Academy of Science in area of computer science, computer engineering and automation. At the moment he holds a position of senior researcher at the Laboratory of Computer Security Problems of SPIIRAS. He is the author of more than 50 refereed publications published in refereed journals and in proceedings of international conferences and symposia, and has a high experience in the research of computer network security. His primary research interests include computer network security, intrusion detection and security visualization.

Igor Kotenko graduated with honors from St.Petersburg Academy of Space Engineering and St. Petersburg Signal Academy. He obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Head of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. He is the author of more than 250 refereed publications, including 12 textbooks and monographs. Igor Kotenko has a high experience in the research on computer security and participated in several projects on developing new security technologies. For example, he was a project leader in the research projects from the US Air Force research department, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. The research results of Igor Kotenko were tested and implemented in more than fifty Russian research and development projects.