Detecting Periodicity in Serial Data through Visualization*

E.N. Argyriou and A. Symvonis

Department of Mathematics,
School of Applied Mathematical & Physical Sciences,
National Technical University of Athens, Greece
{fargyriou,symvonis}@math.ntua.gr

Abstract. Detecting suspicious or malicious user behavior in large networks is an essential task for administrators which requires significant effort due to the huge amount of log data to be processed. However, several of these activities can be rapidly identified since they usually demonstrate periodic behavior. For instance, periodic activities by specific users accessing the billing system of a financial institution may conceal fraud. Detecting periodicity in user behavior not only offers security to the network, but may prevent future malicious activities. In this paper, we present visualization techniques that aim to detect authorized (or unauthorized) user activities that seem to appear at regular time intervals.

1 Introduction

Due to the continuous increase of the size and complexity of computer networks, monitoring the user or network activity in a continuous basis is a necessary and, at the same time a time-consuming task for maintaining the network security. Traditionally, the network monitoring process is achieved by a combination of log file analysis, traffic analysis and intrusion detection systems. Even though most systems are equipped with mechanisms that produce sufficient log files, processing the huge amount of data requires significant effort, and usually is performed with little or no automated support.

Visualization is essential in cases of large data sets such the ones produced in a network, since it interprets the huge amount of data rows into a more comprehensive visual image. The necessity of the visualization aids is due to the fact that it is more difficult to immediately grasp the essence of something, if it is just described in words. In fact, it is hard for the brain to process text. Pictures or images, on the other hand, can be processed extremely well. They can encode a wealth of information and are therefore, well suited to communicate much larger information of data to human. Thus, by taking advantage of the human perception, the analysis of the visualization and the corresponding decision making becomes easier and more efficient. For this reason, over the last few years much research effort has been focused on seeking for visualizations of the network activity that aim to efficiently detect malicious activities.

* The work of E.N. Argyriou has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

G. Bebis et al. (Eds.): ISVC 2012, Part II, LNCS 7432, pp. 295–304, 2012.

The experience of examining malicious events in a network has revealed that many suspicious attempts appear in regular time basis. In several systems such as the billing system of a company, membership renewals systems, etc, periodic events may conceal fraud. For instance, in a billing system, an employee's monthly activity towards a specific customer account is considered to be suspicious, especially if it occurs before the billing day.

Motivated by the fact that detecting periodicity in serial data helps in rapidly identifying suspicious events, we present a system that visualizes serial data (either static or dynamic) produced by systems similar to the ones mentioned above. The main goal is to identify suspicious activities that may consist fraud. In our approach, each event corresponds to a pair of employee-customer due to the nature of the data sets examined. However, this approach can be generalized to other similar systems where appropriately defined pairs of entities can be identified in the system. The proposed system produces different types of visualizations, such that periodic events that are considered to be suspicious are easily identified. In order to produce aesthetically pleasant visualizations that are eventually easy to read and interpret, we employ standard techniques adopted from graph drawing in conjunction with our visualization techniques. As expected from such a system, it is equipped with supplementary functionalities such as support for storing, reloading and post-processing of data. It provides advanced graphic functionality, including popup menus, printing capabilities, custom zoom, fit-in window, selection, dragging and resizing of objects.

The rest of this paper is structured as follows: Section 2 overviews related work. In Section 3, we sketch our contribution. In Section 4, we describe in detail the proposed system. We conclude in Section 5 with open problems and future work.

2 Previous Work

During the last few years various visualization approaches have been proposed for network monitoring. Mansmann et al. [1] presented a visual analytics tool that visualizes the behavior of hosts or higher level network entities over time. Yin et al. [2] presented a novel approach to the visualization of traffic flows to detect and investigate anomalous traffic between a local network and external domains, whose central aspect is a parallel axes view used to represent the origin and destination of network traffic. Shabtai et al. [3] presented two tools that enable the user to visualize and explore time-oriented security data. Lakkaraju et al. [4] presented NVisionIP, a tool that supports different visualizations in order to provide a snapshot of the activity of a network, which supports filtering and aggregation of the input data based on a number of attributes that are important for security analysis. Vandenberghe [5] presented a data visualization tool that analyzes a security event from a range of visual perspectives using different detection algorithms. Finn and North [6] presented a security visualization tool capable of representing tens of thousands of hosts simultaneously and allows the user to display communication patterns between arbitrary locations.

Regarding visualizations of data produced by intrusion detection systems (or IDS for short), Abdullah et al. [7] presented IDS Rainstorm, a tool that provides high-level overviews of intrusion detection alerts. Tolle and Niggemann [8] propose a system

supporting the detection of intrusions and network anomalies by analyzing and visualizing traffic flows in computer networks by means of graph drawing techniques. Oline and Reiners [9] propose several 3-dimensional visualizations, each of which emphasizes on different aspects of IDS alerts. Erbacher et al. [10] presented different techniques for the visual representation, exploration and analysis of IDS related data in order to ease the identification and analysis of network attacks.

Carlis and Konstan [11] presented a spiral visualization technique to highlight a type of data (called it serial periodic), which occurs frequently. According to their approach, serial attributes of the data set are displayed along the spiral axis, while the periodic ones along the radii of the spiral. Weber et al. [12] presented a visualization system for time-series data based on spirals that processes large data sets and detects periodic data patterns. According to their approach, the spiral corresponds to the time axis, while the other attributes of the data are represented by points, colors, lines or bars. Bertini et al. [13] proposed SpiralView, a tool that supports spiral visualizations to monitor network traffic and helps understanding the evolving of network alarms over time. It also provides identification of periodic patterns. An overview on the visualization of time-series data and the available techniques can be found in [14,15,16].

In the context of graph drawing, force-directed methods [17,18,19,20] are quite common when visualizing combinatorial information by means of directed or undirected graphs. In such a framework, a graph is treated as a physical system on which appropriate forces (either attractive or repulsive, or both) are applied. The equilibrium state of the system produces a good configuration or a drawing of the graph. An overview of force-directed methods and their variations can be found in classical graph drawing books [21,22].

3 Our Contribution

Our contribution consists of three different visualization methods that aim to help in identifying periodic activity in time-series data stemming from system involving pairs of entities, e.g., a billing system. The main goal is to contribute in fraud detection. Each visualization results in drawings which can be utilized in order to detect periodicity in the analyzed events. Note that, we measure periodicity by introducing a new metric which is appropriately defined to reveal the frequency in which an event occurs within a time window. Our main visualization method results in drawings consisting of concentric circles whose radius correspond to the periodicity of the activity of each pair of entities of the system. Events that are considered to be suspicious are easily identified since they are dragged towards the center of the circles. Also, a force-directed approach is employed in order to provide a better configuration of the events over the visualization. The system is equipped with supplementary functionalities such as information regarding the entities' activities, examination of certain period of interest or visualization of the series of events.

4 Description of the System

The system's input data sources can be either a log file or a set of records of a database of a system involving pairs of entities, e.g., a billing system. However, it is extensible

to other similar data sources. Each pair of entities is associated with a series of *events* involving them (e.g., a phone call between them, a transaction, etc.), which we assume to be sorted by date. In order to produce a visualization, the system preprocesses these series of events, and estimates a proper period of activity for each pair of entities.

4.1 Periodicity Estimation

For each pair of entities, we introduce a metric that estimates a periodicity value with a specific confidence degree. Let ρ be a pair of entities and n_ρ the number of events associated with pair ρ. Assume that e_i^ρ, e_{i+1}^ρ, $i = 1, \ldots, n_\rho - 1$, are two consecutive events, and let $d_{i,i+1}^\rho$ be their time distance (say measured in days). A time-series $T_\rho = (t_1^\rho, \ldots, t_{n_\rho-1}^\rho)$ is generated by assigning to each event e_i^ρ a value t_i^ρ according to the following formula:

$$t_i^\rho = \begin{cases} \sum_{j=1}^{i-1} d_{j,j+1}^\rho & \text{if } 1 \leq i < n_\rho \\ 0 & \text{if } i = 0 \end{cases}$$

For a given period value s, the ideal time-series $D_s = (0, s, 2s, \ldots)$ is defined by the time stamps that occur if the events between the entities of ρ appear in time intervals that equal exactly to s (see Figure 1). For instance, in case of a period value of 30 days, the ideal time series is $D_{30} = (0, 30, 60, \ldots)$.

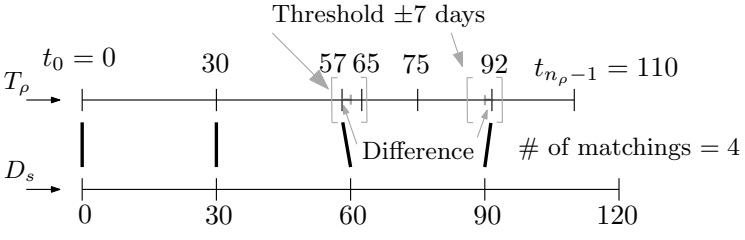

Fig. 1. Line T_ρ corresponds to time-series events of a pair of employee-customer, whereas line T_s' to the ideal time-series for a period of 30 days

Let $t \in T_\rho$ and $\lambda \in D_s$. We say that t and λ *match each other with respect to a threshold value* $\tau \in [1, s/2)$, if it holds that $t \in [\lambda - \tau, \lambda + \tau]$. Let N_ρ^τ be the set of time stamps of ideal time-series D_s, which can be matched with a time-stamp of time-series T_ρ, i.e., $N_\rho^\tau = \{\lambda \in D_s : \exists\, t \in T_\rho \; s.t., \; t \; and \; \lambda \; match\}$. With slight abuse of terminology, we refer to the cardinality of N_ρ^τ as the *number of matchings* of pair ρ. Let also, $diff_\rho^\tau : N_\rho^\tau \to \mathbb{R}$ with:

$$diff_\rho^\tau(\lambda) = \min\{|t - \lambda| : \; t \in T_\rho, \lambda \in N_\rho^\tau \; and \; (t, \lambda) \; match\}$$

The confidence level of a pair ρ of entities with periodicity value s and threshold matching value τ is given by the following formula:

$$confidence(\rho, s, \tau) = \frac{\sum_{\lambda \in N_\rho^\tau} 1 - \frac{diff_\rho^\tau(\lambda)}{\tau}}{|N_\rho^\tau|}$$

Observe that the confidence values belong in $[0, 1]$. Obviously, if the time-series T_ρ is identified with the ideal time-series D_s, then $confidence(\rho, s, \tau) = 1, \forall \tau \in [1, s/2)$. In order to provide a more accurate estimation of the confidence value of a pair ρ for a given period s, with respect to a threshold value τ, one can alternatively compute the confidence value of ρ for all ideal time-series $D_s^i = (i, s + i, \ldots)$, $i = 0, \ldots, s/2$ and, keep the one that maximizes the confidence value.

For a given pair of entities ρ and a prespecified threshold value τ, we measure its confidence for all periodicity values $s \in [1, s_{max}]$, where s_{max} corresponds to the maximum periodicity value defined by the user. Having determined all confidence values of ρ, the *periodicity* of pair ρ (with respect to the specified threshold value τ), equals to the periodicity value that maximizes its confidence value.

Note that for each pair of entities, the system is able to produce a visualization similar to the one of Figure 1, in order to present the series of events associated with the specific pair and the matchings with the ideal time-series. In addition, the system is capable of identifying weekends and feast days of each year, and adapts appropriately the ideal sequences. However, for simplicity reasons, in our description we ignored this functionality.

4.2 Periodicity Visualization

The main visualization of the system is illustrated in Figure 2, where we seek for monthly periodic activity. It consists of a system of concentric circles whose radius correspond to different periodicity values. The nodes of the visualization correspond to pairs of entities. The outermost circle corresponds to a period of 8 days, while the innermost to 31. We only compute periodicity values that are greater than the threshold value, which in the visualization of Figure 2 is set to 7 days. However, this is a value determined by the user. With this configuration, nodes with periodicity of 30 or 31 days are dragged towards the center of the system.

The system is also split in circular sectors that correspond to different number of matchings with the ideal time-series for each period, as discussed above. For a more uniform arrangement of the nodes in the system of circles, nodes whose number of matchings is greater than the median value lie on the upper semicircle, while the remaining ones at the bottom. The maximum number of matchings corresponds to the midpoint of each upper semicircle. In the visualization, we ignore nodes whose time-series had up to two matchings with the ideal time-series since otherwise, we always have a perfect sequence of value two. The gray colored areas of Figure 2 illustrate nodes that appear to have suspicious activity (due to their periodicity values) and need to be further examined. We have also chosen to highlight the entire ring of periods greater than 27, even in cases with few matchings, since this may reveal a suspicious behavior that is about to start.

The system provides the capability to the user to select a node (especially a suspicious one) and draws with the same color or shape all nodes of the system that contain

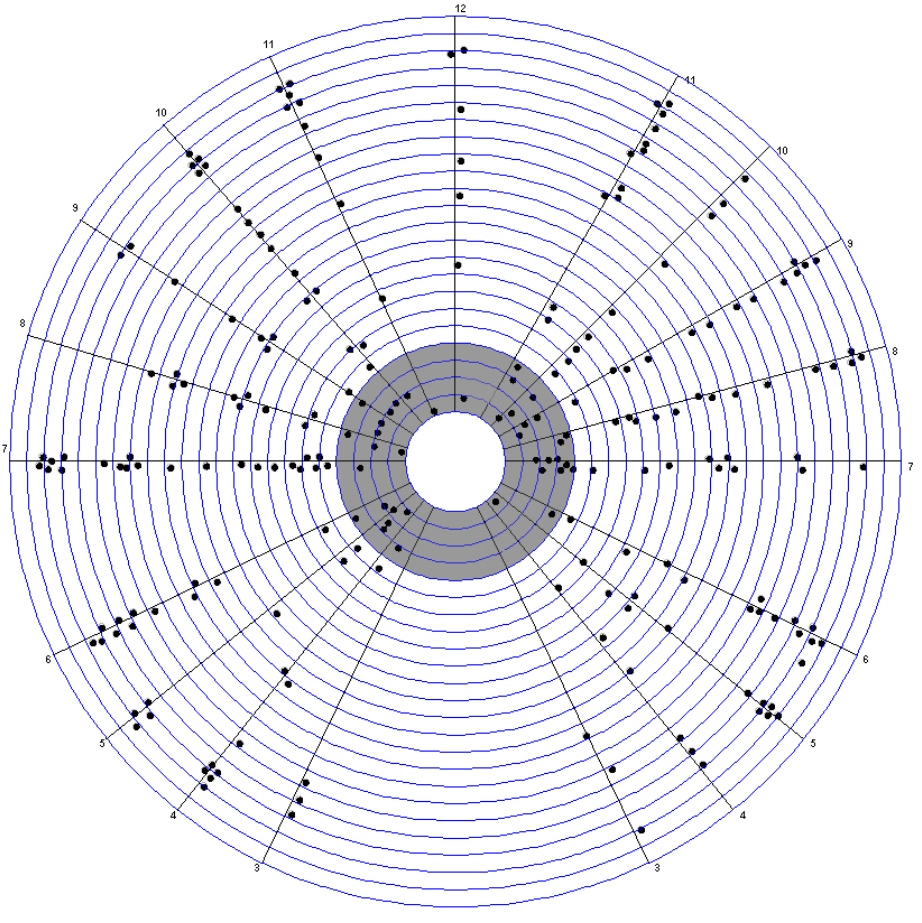

Fig. 2. A concentric circle system in which each radius correspond to a periodicity value. The gray-colored areas are the ones that have to be examined first for suspicious activities.

the same entity with the selected one. In this manner, the user can identify whether the entity appears to have a continuous suspicious activity. Also, the system can draw with different colors or shapes the most suspicious nodes, such that they can be easily distinguished from the remaining ones, as in Figure 3. The system is equipped with popup menus at each node which reveal additional information, such as periodicity, confidence value and so on. The system also provides supplementary functionalities such as support for storing, reloading and post-processing of the visualization. It also supports advanced graphic functionality, including printing capabilities, custom zoom, fit-in window, selection, dragging and resizing of objects.

In order to obtain more legible drawings, we have used the classical force-directed algorithm of Eades [18] in conjunction with our visualization technique. A force-directed algorithm models the vertices of the graph as electrically charged particles that repel each other, and its edges by springs in order to attract adjacent vertices. However,

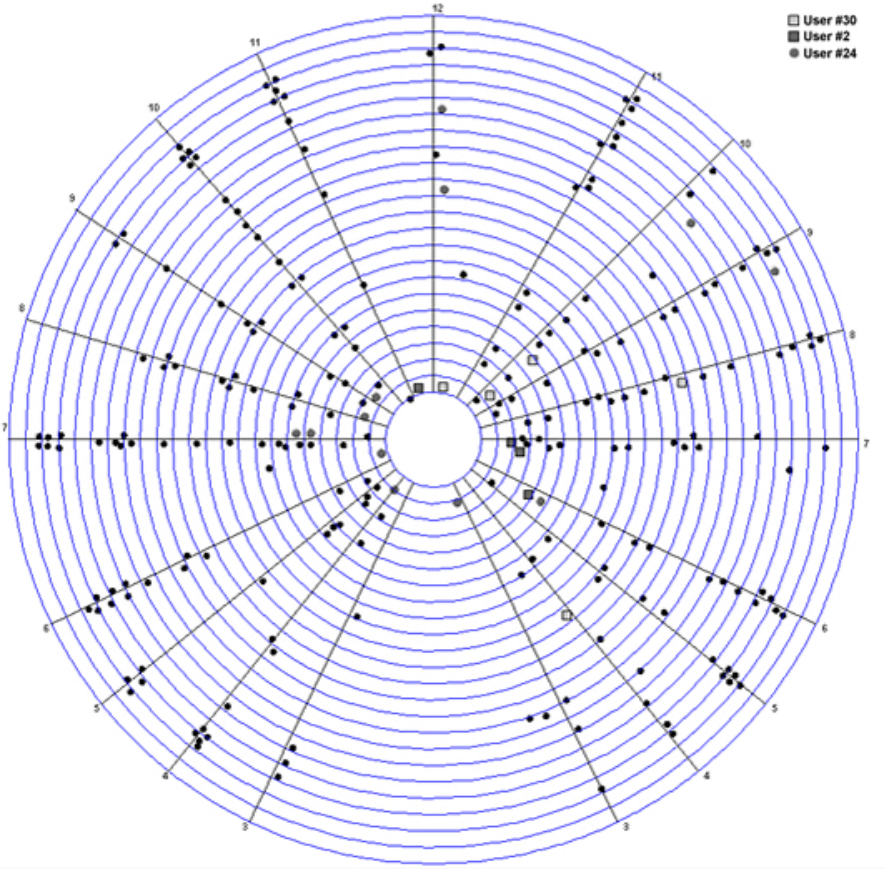

Fig. 3. The top three most dangerous entities are illustrated with different colors and shapes

before we proceed with the detailed description of the algorithm, we introduce some necessary notation. Let $G = (V, E)$ be an undirected graph. Given a drawing $\Gamma(G)$ of G, we denote by $p_u = (x_u, y_u)$ the position of node $u \in V$ on the plane. The unit length vector from p_u to p_v is denoted, by $\overrightarrow{p_u p_v}$, where $u, v \in V$.

In our approach, we add dummy nodes on each circle and along the lines that splits these circles in circular sectors. Each dummy node corresponds to the number of matchings for a given periodicity. Then, we use springs to connect each node with the dummy node of its period circle that corresponds to its number of matchings. The springs follow the logarithmic law instead of the Hooke's law, in order to avoid exerting strong forces on distant nodes. The attractive forces follow the formula:

$$\mathcal{F}_{spring}(p_u, p_v) = C \cdot \log \frac{||p_u - p_v||}{\ell} \cdot \overrightarrow{p_u p_v}, \ (u, v) \in E$$

where C and ℓ capture the *stiffness* and the *natural length* of the springs, respectively. We also, use repulsive forces among the nodes of the visualization, in order to avoid node overlaps. The repulsive forces are defined as follows:

$$\mathcal{F}_{rep}(p_u, p_v) = \frac{C_p}{||p_u - p_v||^2} \cdot \overrightarrow{p_u p_v},\ u, v \in V$$

where C_p is a *repulsion* constant. The set of forces that were described assure that in an equilibrium state of the model, the nodes will be eventually drawn close to their associated periodicity circles and more precisely, close to the dummy nodes that "describe" their number of matchings. Note that, we do not apply forces on the dummy nodes. Hence, their positions remain unchanged.

4.3 Single Period Visualization

In order to have a better insight of the nodes that lie on a specific period ring (e.g., when examining activities in a period of 30 days), the system is capable of producing a visualization with concentric circles (similar to the one mentioned above; see Figure 4) that contains nodes of a specific period value. In this case, the radii of the concentric circles correspond to different degrees of confidence. The outermost circle corresponds to confidence value 0.1, while the innermost to 1. Hence, nodes for which the confidence value tends to 1 lie towards the center of the system. As above, the visualization is split in circular sectors based on the estimated number of matchings and simultaneously supports all functionalities of the previous visualization. Again, the final layout is computed using a force-directed algorithm that is a simple variation of the one described in Section 4.2.

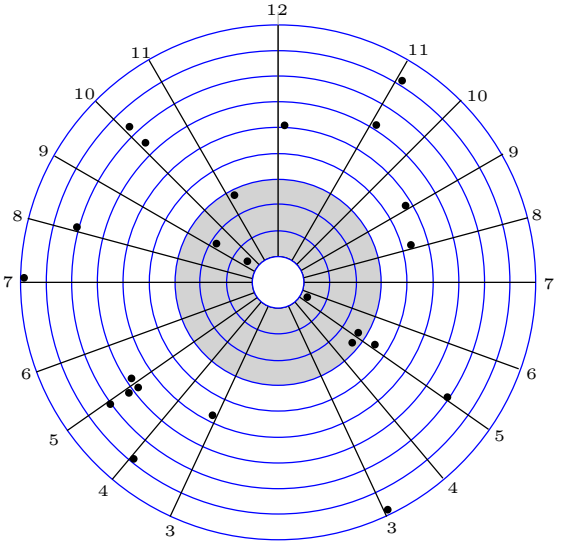

Fig. 4. A concentric circle visualization for a period of 30 days. The radii of each circle correspond to different confidence values. Nodes with confidence value 1 move towards the center of the system.

5 Conclusions and Future Work

In this paper, we presented a system that aims to detect periodic event in time-series data. The system is oriented towards fraud detection in data stemming from billing or other similar business systems. However, it can be extended to support data from other data sources. The presented visualizations help the security managers to identify employees that appear to have suspicious activity towards specific customer accounts. Of course, our work opens several directions for future work:

- One of the main future goals of this system is to be enhanced with several other visualizations methods that reveal periodicity. More sophisticated algorithms adopted from Graph Drawing or Information Visualization need to be incorporated.
- Alternative metrics to measure the confidence degree can be used in order to obtain more accurate periodicity estimations. This may also affect the quality or the type of the produced visualizations.
- Identifying group of users (instead of a particular user) that appear to have similar suspicious behavior is also of interest. Standard clustering techniques adopted from Graph Drawing may be useful for the production of such visualizations.
- Incorporating more functionalities required for a security manager such as statistic analysis of the activity for each entity, plots, bar charts, etc.

Acknowledgements. We would like to thank Vassilis Vassiliou for his useful suggestions and comments related to fraud detection.

References

1. Mansman, F., Meier, L., Keim, D.A.: Visualization of host behavior for network security. In: VizSEC 2007, pp. 187–202. Springer, Heidelberg (2008)
2. Yin, X., Yurcik, W., Treaster, M., Li, Y., Lakkaraju, K.: Visflowconnect: netflow visualizations of link relationships for security situational awareness. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC 2004, pp. 26–34. ACM, New York (2004)
3. Shabtai, A., Klimov, D., Shahar, Y., Elovici, Y.: An intelligent, interactive tool for exploration and visualization of time-oriented security data. In: Proceedings of the 3rd International Workshop on Visualization for Computer Security, VizSEC 2006, pp. 15–22. ACM (2006)
4. Lakkaraju, K., Yurcik, W., Lee, A.J.: Nvisionip: netflow visualizations of system state for security situational awareness. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC 2004, pp. 65–72. ACM (2004)
5. Vandenberghe, G.: Network Traffic Exploration Application: A Tool to Assess, Visualize, and Analyze Network Security Events. In: Goodall, J.R., Conti, G., Ma, K.-L. (eds.) VizSec 2008. LNCS, vol. 5210, pp. 181–196. Springer, Heidelberg (2008)
6. Fink, G.A., North, C.: Root polar layout of internet address data for security administration. In: Proceedings of the IEEE Workshops on Visualization for Computer Security, VIZSEC 2005, pp. 55–64. IEEE Computer Society (2005)
7. Abdullah, K., Lee, C., Conti, G., Copeland, J.A., Stasko, J.: Ids rainstorm: Visualizing ids alarms. In: Proceedings of the IEEE Workshops on Visualization for Computer Security, VIZSEC 2005, pp. 1–10. IEEE Computer Society (2005)

8. Toelle, J., Niggemann, O.: Supporting intrusion detection by graph clustering and graph drawing. In. In: Proc. of 3rd Int. Workshop on Recent Advances in Intrusion Detection, RAID 2000 (2005)
9. Oline, A., Reiners, D.: Exploring three-dimensional visualization for intrusion detection. In: Proceedings of the IEEE Workshops on Visualization for Computer Security, VIZSEC 2005, pp. 113–120. IEEE Computer Society (2005)
10. Erbacher, R.F., Christensen, K., Sundberg, A.: Designing visualization capabilities for ids challenges. In: Proceedings of the IEEE Workshops on Visualization for Computer Security, VIZSEC 2005, pp. 121–127. IEEE Computer Society (2005)
11. Carlis, J.V., Konstan, J.A.: Interactive visualization of serial periodic data. In: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST 1998, pp. 29–38. ACM (1998)
12. Weber, M., Alexa, M., Müller, W.: Visualizing time-series on spirals. In: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS 2001), pp. 7–14 (2001)
13. Bertini, E., Hertzog, P., Lalanne, D.: Spiralview: Towards security policies assessment through visual correlation of network resources with evolution of alarms. In: Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology, VAST 2007, pp. 139–146. IEEE Computer Society (2007)
14. Silva, S.F., Catarci, T.: Visualization of linear time-oriented data: A survey. In: Proceedings of the First International Conference on Web Information Systems Engineering (WISE 2000), vol. 1, pp. 310–319. IEEE Computer Society (2000)
15. Müller, W., Schumann, H.: Visualization for modeling and simulation: visualization methods for time-dependent data - an overview. In: Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, WSC 2003, pp. 737–745 (2003)
16. Aigner, W., Bertone, A., Miksch, S., Tominski, C., Schumann, H.: Towards a conceptual framework for visual analytics of time and time-oriented data. In: Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best is Yet to Come, WSC 2007, pp. 721–729 (2007)
17. Davidson, R., Harel, D.: Drawing graphs nicely using simulated annealing. ACM Transactions on Graphics 15, 301–331 (1996)
18. Eades, P.: A heuristic for graph drawing. Congressus Numerantium 42, 149–160 (1984)
19. Fruchterman, T., Reingold, E.M.: Graph drawing by force-directed placement. Software-Practice and Experience 21, 1129–1164 (1991)
20. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters 31, 7–15 (1989)
21. Kaufmann, M., Wagner, D. (eds.): Drawing Graphs. LNCS, vol. 2025. Springer, Heidelberg (2001)
22. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall (1999)