SpiralView: Towards Security Policies Assessment through Visual Correlation of Network Resources with Evolution of Alarms

Enrico Bertini* University of Fribourg
Patrick Hertzog† NEXThink S.A.
Denis Lalanne‡ University of Fribourg

ABSTRACT

This article presents *SpiralView*, a visualization tool for helping system administrators to assess network policies. The tool is meant to be a complementary support to the routine activity of network monitoring, enabling a retrospective view on the alarms generated during and extended period of time. The tool permits to reason about how alarms distribute over time and how they correlate with network resources (e.g., users, IPs, applications, etc.), supporting the analysts in understanding how the network evolves and thus in devising new security policies for the future. The spiral visualization plots alarms in time, and, coupled with interactive bar charts and a users/applications graph view, is used to present network data and perform queries. The user is able to segment the data in meaningful subsets, zoom on specific related information, and inspect for relationships between alarms, users, and applications. In designing the visualizations and their interaction, and through tests with security experts, several ameliorations over the standard techniques have been provided.

Keywords: Network security, Intrusion Detection, Visualization, Data Exploration

Index Terms: C.2.0 [Computer-Communications Networks]: Security and protection— [C.2.3]: Computer-Communications Networks—Network management H.5.2 [User Interfaces]: Graphical user interfaces (GUI), Interaction Styles—

1 INTRODUCTION

Network security is a fervent branch of research and a successful business area devoted to monitoring, analysis, and understanding of network data to improve security and efficiency. Network administrators typically use a mix of tools, e.g., error logs, traffic flow reports, intrusion detection systems (IDS), to secure the network over attacks and misuse, and to react fast in case of detected dangerous behaviors. Traditional tools like IDS systems, which continuously monitor the network, either matching the current traffic with known attack patterns (signature-based IDS) or trying to detect potentially dangerous behaviors (anomaly-based IDS), report their data in textual logs that the administrator must sort out to make sense of the network's state. While this is common practice, and proved to be, to some extent, successful, there is a growing interests and need in using visualization to make the process more efficient, effective, and easier to perform.

Visualization is a promising solution because it exploits the parallel processing capabilities of the human visual system and thus it allows for analyzing large quantities of data at a glance [10]. Moreover, the tight relationship between interaction and visualization permits to reason about information, allowing an analyst to learn from exploration and to spot unexpected trends.

*enrico.bertini@unifr.ch
†patrick.hertzog@nexthink.com
‡denis.lalanne@unifr.ch

IEEE Symposium on Visual Analytics Science and Technology 2007
October 30 - November 1, Sacramento, CA, USA
978-1-4244-1659-2/07/$25.00 ©2007 IEEE

To date, there already exist several visualization tools for network monitoring. Current prototypes focus on visualizing (in near real time) either network traffic [25][14][6][9] or network alarms [1] [13] [8], rarely a mix of them, and to place them in a context that helps in understanding their nature. Some of them use raw data coming from traffic logs, some others employ preprocessed/digested data. All these systems share the common goal to make the task of network monitoring more efficient and effective. They mainly support the user in building a consistent picture of the whole network state without requiring to mentally join pieces of information scattered around many applications and files. In turn, they help the administrator to feel in more control and to achieve what is commonly known as "situational awareness".

In this paper we present *SpiralView*, a tool designed to meet a related but different purpose. While it can also be used to monitor the network, its primary purpose is to support the analysts in reasoning about how the network evolves and in taking informed decisions on how to administrate it. The focus is shifted from day-to-day monitoring, as a way to spot dangerous events and react, to the analysis of extended periods of time to devise policies that improve the network's behavior. Examples include: better targeted awareness programs, restriction or relaxation of network constraints, redefinition of access rules.

Similarly to most recent systems, the system visualizes the alarms coming from an analysis engine and permits to correlate them with the available network resources. We use a proprietary engine developed by NEXThink S.A. [1] which, differently from existing engines, is able to convey, other than traditional data like IPs, ports, etc., information about applications and users. In Section 3 additional details about this solution and the network data will be provided. Our tool permits to analyze alarms across extended time periods (days, weeks, months) and to segment them according to their distribution over multiple network resources and parameters (e.g., alarm types, users, applications). The patterns found can be used to tune some of the engine's parameters and to observe the consequent network's behavior. To this end, the system also allows to attach notes to alarms or specific moments in time to remember when some strategies have been implemented.

The system is organized around a spiral visualization, representing linear evolution of time, into which alarms are laid out using the time of their first appearance in the network. The spiral's circular and sequential behavior allows to follow temporal evolution and to detect periodic trends at the same time [5][23]. We are, in fact, interested in observing how alarms evolve in time and, at the same time, if any prominent periodic patterns exist (e.g., alarms appearing everyday at the same time). The tool is provided with additional views, coupled with the spiral, to visualize network resources and attributes. Their design is based on simple interactive bar charts and a custom user/application view which we chose because of their familiarity and ease of use. The bar charts measure the number of alarms falling in each category. As an example, the top bar chart in Figure 1 presents the number of alarms pertaining to alarm's type category (e.g., network scanning, malicious activities, etc.). The user/application view is very similar in spirit to the bar

[1] http://www.nexthink.com/

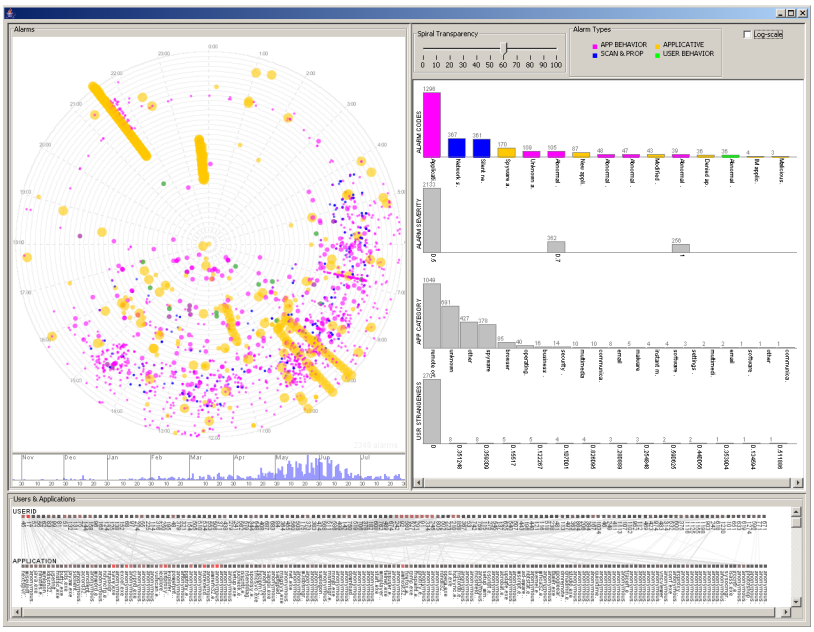

Figure 1: The whole SpiralView's interface.

charts in that it provides the same interactive capabilities but uses color instead of bar heights to convey information; a useful design feature that permits to visualize many resources at once. Additional details will be provided in Section 4.2.1. The implemented interaction is a key feature of the system. The user can select single or a combination of bars, similarly to *brushing histograms* [15][17], to make queries and filter out alarms that are not within specified categories. The tool implements a two-way interaction mechanism. A selections performed on the bar charts filters out and thus segment the set of alarms, a selection of alarms in the spiral permits to select groups of interest and see how they map onto the network resources. An in-depth description of our design choices in developing the visualizations and their interaction is provided in Section 4.

In summary our original contribution consists of the following points:

- The focus is shifted from day-to-day monitoring to long term analysis and monitoring of anomalous activities.
- Information about *users* and the *applications* they use are provided, thus permitting to abstract related network details like IPs, ports, etc.
- Commonly used visual techniques like spiral and bar charts are customized, enhanced, and integrated into an original user interface to provide powerful interactive exploration tools.

Section 2 provides background information and related work. Section 3 describes the network data we use and the underlying system. In Sections 4 and 5 the design of *SpiralView* is introduced through illustrative examples and case studies. Finally, in Section 6, 7, and 8, the article presents the evaluation phase, the future research challenges we propose to tackle, and the conclusions.

2 Background and Related Work

Alarms are used for numerous purposes such as safety alarms (earthquakes, fire, tornado, nuclear power plants, etc.), burglar alarms designed to warn of intrusions, clock alarms (meeting, task, etc.) or network security. With any kind of alarm, the need exists to balance between on the one hand the danger of false positive alarms and on the other hand failing to signal an actual problem, i.e. false negative. Both can be dangerous, wasting energy, acclimatizing people to ignore alarm signals, or missing emergencies. For these reason alarm management is critical and the inclusion of human factors into alarm systems design necessary to improve usability and thus security. The major usability problem generally relies on the fact that there are too many alarms, commonly referred to as alarm flood. However, there can also be other problems with an alarm system such as poorly designed alarms, improperly set alarm points, ineffective annunciation, unclear alarm messages, etc. [18]

Visualization techniques seem particularly suited to solve the problem of alarms management. Unfortunately there are very few attempts that make profit of these techniques to decrease the cognitive load of operators and at the same time increase their understanding of the causes.

In the network security domain, which is of particular interest in the work presented in this article, several tools based on information visualization have appeared during the last years; they can be divided in different groups based on their information source. Tools such as visFlowConnect [25], nVisionIP [14], RUMINT [6] or TNV [9] use network flows or packet inspection. Applications like MieLog [19] or Tudumi [20] use logs collected directly on the endpoints. On the other hand, RainStorm [1], SnortView [13], STARMINE [12], VisAlert [8], or other visualization-based tools that use an hybrid approach [11], visualize alarms generated by traditional security systems (e.g, IDS). This last kind of systems are the most similar to ours in that they visually correlate alarms and network resources. All of them, however, are designed to visualize the events of a limited time period (e.g., the last 24 hours) and are thus limited to day-to-day monitoring. *SpiralView* extends this paradigm and allows for both monitoring and analyzing alarms on a much longer time span. In addition, our tool, thanks to the proprietary engine utilized, can access some additional and more comprehensible data like network applications and users.

Since our primary visualization technique for alarms is a spiral, it is worth to mention that the same technique has been successfully used in a number of other visualizations, both for general time based data [5][23][21] and in the specific field of network security

[16][7]. None of these however uses a spiral as a component of a larger system, as in our case. The interaction between the different views and some custom enhancements of the technique we implemented in our tool can be of interest for other purposes.

3 Security system's architecture and data

In order to better understand the role of the *SpiralView* and its added value, it is useful to briefly describe the environment in which it takes place. *SpiralView* is an ongoing research effort integrating into the well established NEXThink's suite that is made of three main elements, as depicted in Figure 2:

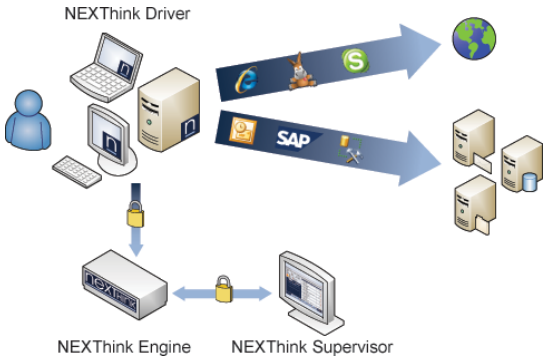

Figure 2: NEXThink suite architecture

- Driver - Runs passively at the endpoint devices (and transparently from users and applications) capturing real time footprints of networked application activities. These footprints carry information about application binaries (hash and version) as well as identification of the associated user account. Every single network activity (connection) is analyzed within the context of a certain user account using a specific application.

- Engine - Is the core analysis engine that contains a patented new artificial intelligence belief detection technology [2]. This technology is mainly composed of belief networks which efficiently model user and application behaviors, detecting deviations and triggering alarms to notify about security threats.

- Supervisor - Provides powerful tools for day-to-day monitoring and investigation of network events through advanced visualizations and effective user interaction.

SpiralView brings to the solution a way to analyze alarms and their implications with a long term perspective. If one detects, for instance, an abnormal behavior of a group of users, one could take different possible measures ranging from technical ones (e.g., new firewall rules) to educational ones (e.g., awareness program). With the long term analysis provided by *SpiralView*, one is able to assess if those measures have been successful or if new ones are necessary.

SpiralView is also very useful to improve the detection performed by the analysis engine. With the display of alarms over a long time period, one can detect events that are not really pertinent for the security of the network. One can then give that feedback to the engine in the form of tunings. Tunings can alter the different detection algorithms to make them fit better in the particular network environment. Therefore, we have an iterative process between *SpiralView* and the analysis engine that enables us to improve both the detection and the visualization. This loop between the visualization system and the engine, cross enhancing each other, makes the whole environment an effective visual analytics solution.

[2]The technology is the result of research carried out at the Artificial Intelligence Laboratory at the Swiss Federal Institute in Lausanne (EPFL)

3.1 Network alarms and their contextual information

Current systems are based either on logs (computer logs or network flows) or on alarms generated by traditional security systems. In both cases, collected information is made of a high number of low-level parameters that are difficult to interpret by the security administrator: she/he has to do a lot of inferences to transform those low-level parameters into information useful to take a decision. This lengthens considerably the process and moreover it introduces some incertitude because assumptions have to be done. For instance, one can have a HTTP flow on port 80 with some headers indicating that it may be *Internet Explorer* but no one can be sure that it is not a malicious application trying to pass itself off as *Internet Explorer*.

Our approach is to obtain only a small set of certain high-level parameters for each connection on the network.

Definition 1 *A* Connection *is the tuple* <time, user, source host, application, destination port, destination host> *where*

- time *is the time stamp at the beginning of the connection,*
- user *is the user who triggered the connection* (e.g., *the user can be identified by the SID on Windows or the UID on UNIX-based operating systems),*
- application *is the application (we mean the binary,* e.g., *firefox.exe, not the protocol,* e.g., *HTTP) used to initiate the connection,*
- port *is the port used on the target computer and*
- source / destination hosts *are the identification of the source and destination computers of the connection.*

Those parameters are collected on the endpoints and centralized on the network where they are correlated and analyzed by an engine based on different artificial intelligence techniques to produce, if necessary, comprehensive alarms.

Definition 2 *An* Alarm *is the tuple* <time, code, severity, {connections}> *where*

- time *is the time stamp when the analysis engine created the alarm,*
- code *represents the type of the generated alarm,*
- severity *gives a qualitative estimation the potential harm of the incriminated connections and*
- {connections} *is the list of connections, as described in Definition 1, that triggered the alarm.*

In the following we refer to *network resources* as a generic term to indicate all kinds of information that can be extracted from the alarms. Thus with this term we indicate both network elements such as applications and metrics (e.g., severity).

4 The SpiralView: visual and interaction design

As mentioned in the introduction, the idea of developing a visualization of alarms started with the analysis of visual interfaces provided by current systems. One limitation of current systems is that they provide a limited view on the set of alarms generated, preventing the administrator to draw a big picture of the network and of its evolution in time; both in terms of its performance and security. A better overview on alarms, with the possibility of accessing key

information related to them, is largely desirable. With such a view the administrator might increase his knowledge on how the alarms distribute and evolve over time and permit to evaluate the effect of network policies. Based on the analysis of what happened in the past, one can monitor the evolution of the system when a new policy is entered.

With these requirements in mind, we have developed a composite tool depicting the whole set of alarms in a spiral visualization and providing interactive inspection tools to focus on interesting alarms and extract relevant information connected to them. The user interface can be split into two logical areas where different types of information are displayed. The spiral is in charge of representing the evolution of alarms in time, the rest is in charge of displaying network resources and metrics. An annotation mechanism is also implemented to permit events and alarms' tracking. In the following we describe the design of each module and how the user can interact with them.

4.1 Alarms Visualization

The spiral axis represents a time-based substrate on which alarms are positioned using their time of appearance in the network. All the alarms generated in the system in the last k months are displayed, staring from the older in the center up to the newer alarms in the outer ring. The spiral shape has the following advantages over other time-based visualizations: 1) it can present data sequentially; 2) it exposes periodic behavior through radial alignments of objects; 3) it assigns more space to recent alarms.

The perception of time periods in the spiral is extremely important. We decided to use a daily period (that is, one ring represents one day) because this exposes the most important periodical pattern to system administrators. Network alarms, in fact, tend to be clustered around specific times in the day. The spiral thus follows a 24 hours period, starting at midnight in the top, following with 6am in the right, noon in the bottom, 6pm in the left. In order to make the time substrate visible and easily put the elements in a time context, the visualization presents one radius per hour and one ring per week (see Figure 1).

The color of alarms represents alarm type, because it is the most important information administrators use to discriminate between alarms, and corresponds to the same colors displayed in the bar charts for a ready correlation. Their size is mapped to alarm severity that is the second most important information. Looking at the spiral it is thus easy to focus on the most important dimensions: alarm type and severity.

The spiral is also coupled with a time histogram at its bottom, which is used to convey aggregate data about how the total number of alarms evolves in time. The histogram is also used to select a time period in the spiral and zoom on it. It's worth to note that this latter function is particularly useful since the selection of time periods on a spiral is rather complicated. With the histogram, instead, the user can easily select time segments to zoom into.

As for zooming, we have implemented an animated zoom that supports the user in the understanding of the view change [24]. When zooming in, each alarm is moved along a radial path and the substrate changes (e.g., the distance between rings grows) to reflect the change in time resolution. Figure 3 shows three progressive zoom views. We tested several strategies for the animation and we found that the radial motion is the one that best preserves temporal coherence and context. As an example, we also tried to animate alarms as if they would rotate around the spiral, up to their end position. The result was quite poor, because the path between the original and the end position can hardly be followed.

A drawback of the spiral occurs when small alarms are hidden behind big alarms, i.e. most severe ones. In order to reduce this negative effect we provide a user-adjustable transparency control that permits to detect hidden objects. It's worth to note however that the problem of overlapping is also reduced by filtering that is the key interactive feature in the system.

Some additional interactive features related to the spiral and the rest of the environment will be described in the following sections where interaction is explained in more detail.

4.2 Resources Visualization

4.2.1 User/Application View

The resources visualizations are split into two main views: 1) a bar chart view containing high level categories and aggregate metrics; 2) a custom users and applications view. The final design is the result of several iterations and walkthrough evaluations with security experts. In designing these views we had to find a synthesis between visualizing categories of elements (e.g., alarm type, application type, user strangeness) and the elements themselves, that is, users and applications. Categories are extremely important to easily segment the alarms in meaningful subsets. But, at the same time having users and applications always visible is a key feature to help administrators forming a mental map and, given their importance in interpreting the network, favoring their recognition rather than their recall.

4.2.2 Interactive Bar Charts

The bar charts count the number of alarms falling in each category and are ordered in frequency order to give an idea about where the majority of alarms fall. When one or more filters are activated each bar shows a "sub-histogram" indicating proportion of elements that fall in the category in the current query. If a bar becomes empty, because for the given query no alarms are in its category, its label is grayed out. This gives a strong visual indication of correlation between dimensions. Since the distribution of elements is often skewed toward some specific values, the elements at the lower end of the scale become too small to be perceived. To overcome this problem the user can activate a log-scale mapping making the small elements more visible.

The user/application view at the bottom of the screen stems from the need of making users and applications always visible. Its behavior is similar to the bar charts in that: 1) its appearance changes when filters are activated, to reflect quantitative information about selected subsets and to gray-out empty elements; 2) the labels can be used as query tools to focus on specific instances. Quantitative information here is conveyed by color instead of bar height. We used a continuously increasing luminance color scale (as required when mapping numerical data to color [22][4]) ranging between black and red. This permits to save a good amount of space which is thus devoted to labels. Colors are updated when alarms are filtered out in order to always reflect the current distribution of alarms. The red shades help to single out the most important values. This view also uses lines between users and applications to directly expose correlation between them. This last feature is crucial to understand which applications are used by which users. Its usefulness becomes clear when filters are activated. Figure 4, where only the alarms classified as *network scanning* are visualized, makes it clear. There are two different clusters of users and applications that generate the majority of network scans. Without these lines the same information could not be easily extracted.

The *SpiralView* presents a two-ways interaction paradigm, from alarms to resources and vice versa, which is realized through a tight integration of *dynamic filtering* [2] and *brushing* [3]. The ideas is developed after the common *information visualization mantra*: "overview first, zoom and filter, details on demand". The user can apply queries on the alarms selecting the dimensions of interest in the bar chart and in the user/application view. The values can be selected clicking on the labels in any disjoint combinations between the same dimension or across different dimensions. As an example the user can decide to see only alarm types: "spyware application"

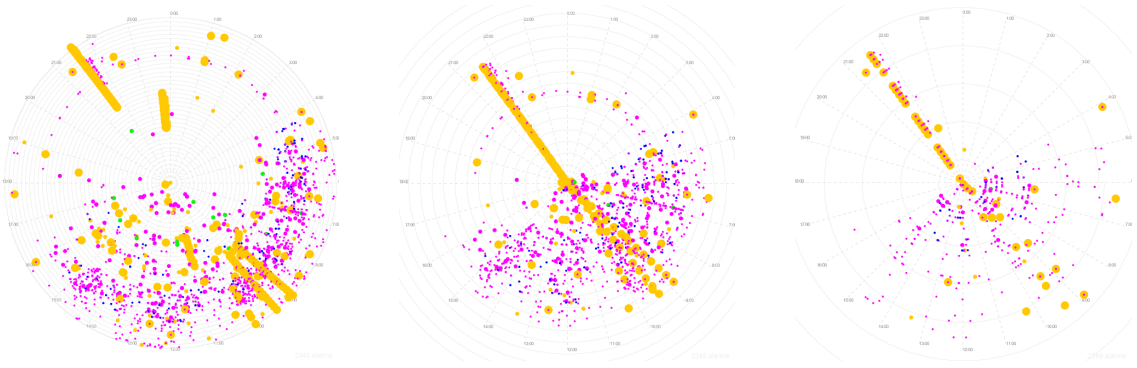

Figure 3: Progressive **radial zoom** sequence in the spiral.

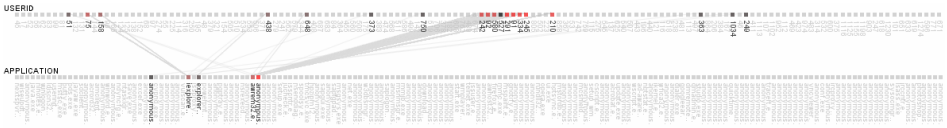

Figure 4: The user/application view. When some elements are filtered-out, the correlation between users and applications stands out.

and "new application" (i.e., disjoint values within the same dimension) or also alarm type "abnormal user behavior" *and* application category "remote control" (i.e., values across different data dimensions).

When the user focuses on a subset of interest, he/she can probe the remaining alarms on the spiral and select some subgroups to see how they map on the user/application view. This is extremely useful because once the spiral visualization is offloaded and few elements of interest remain, the necessary missing information is how these remaining alarms map to the users' population and the applications they use.

These interactive aspects are further explored in the next section where a whole case study is presented. In our opinion, the usefulness and power of the devised interactive tools is emphasized by the explanation of real examples.

4.3 Annotations

Thanks to annotation capabilities the spiral serves also as a communication tool between administrators and as a tool to keep track of the manual interventions made on the network. Indeed, it can be annotated by the analyst in order to label alarms or specific times to measure the effectiveness of deliberate interventions.

For instance, Figure 5 illustrates an annotation entered on the fly by an administrator, explaining the origin of the group of alarms highlighted and also marking the action applied on the engine to relax this type of alarms. This capability is extremely important in that it permits to remember when certain actions take place and thus to compare the status of the system before and after an intervention. Since the primary purpose of the system is to permit long term analysis and policies' assessment, with annotations not only it is possible to devise new strategies but also, to some extent, to check if and how new rules have changed the network's behavior and to share this knowledge between stakeholders.

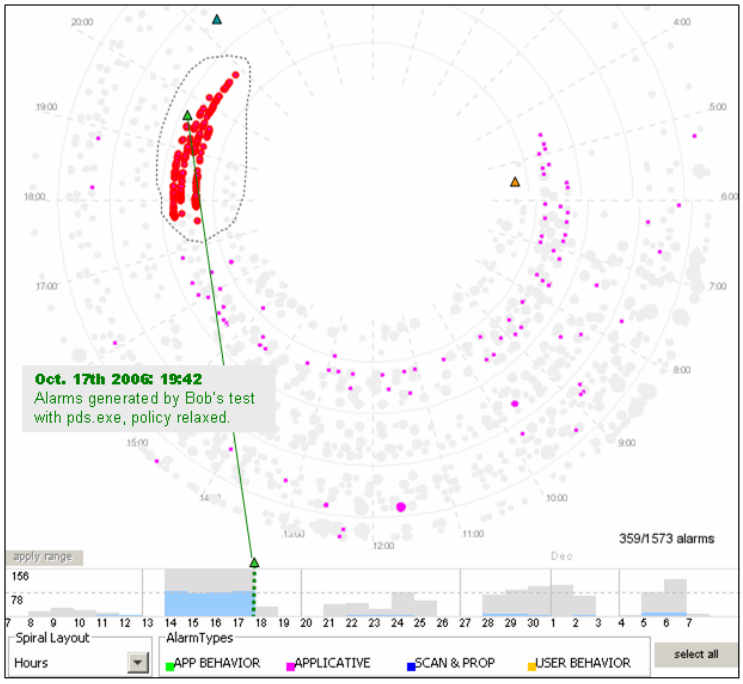

Figure 5: Annotations: the spiral (and the time histogram) can be annotated to keep track of events and to share notes about them.

5 Case Studies

In this section we present two real-world examples to describe the *SpiralView* capabilities in a more integrated fashion and to give an idea about what kind of information can be extracted with it.

The typical usage of this tool occurs some weeks after a new policy has been taken by the administrator, either a new rule or a

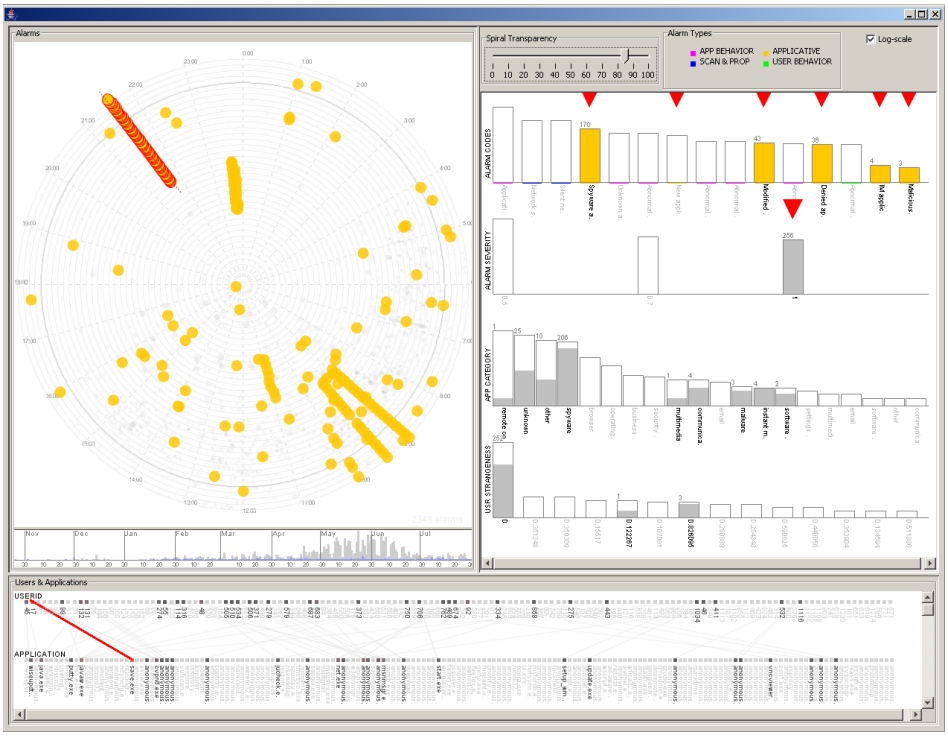

Figure 6: Case Study 1: the alarms generating the ”spikes” in the spiral (alarms recurring everyday, for a long time period, at the same time) are investigated in detail. Through filtering and selection we are able to isolate the source of these recurring alarms. The alarms are generated by the user/application pair: user ”17” and application ”save.exe”.

training program, that needs to be assessed.

Figure 1 shows the whole set of alarms for a time period of nine months. One of the first visible trends in the spiral is a series of alarms that form a sort of spikes: one around 9 o’clock, a second one around 10 and a third one between 21 and 22 o’clock. The spiral shows that these alarms have been recurring for several weeks, since they span several rings. Further, they seems to be all of the same type (same yellow color) and same severity (same size). They are of obvious interest since they have been recurring at the same time in the day for such a long time. In order to draw more information about them, we select on the bar chart all the alarms in the category ”applicative alarms” (the yellow ones) and focus on those with high severity (third bar in the second bar chart) (see Figure 6). This filtering decongests the spiral and thus permits to better understand the alarms under inspection. The user/application view presents several users and applications related to the selected subset. Nonetheless, the pair *user* ”17” and *application* ”save.exe” present a very high red shade, revealing that a large part of the alarms visualized refers to them. To push the analysis a bit further the spikes are selected on the spiral to see how they map to the user/application view. In Figure 6 the relationship becomes crystal clear: all the alarms of the selected spike belong to the same application/user pair, and the same result is obtained probing the other spikes. It represents a series of not dangerous automatic procedures repeating every day at the same time. The administrator decided to turn off the detection of this specific kind of alarms in order to avoid annoying and not useful alarm and to decongest the visualization in future analyses.

A second example is provided in Figure 7, where the analysis starts from the need to inspect how the alarms of type ”scan and propagation” (blue alarms) are distributed along the user population and their resources. Blue alarms have been selected in the bar chart view. As one can see, these alarms have started appearing since the last eight weeks and they cover a period of time during the day between 4 and 16 o’clock without any evident clustering or alignment around some specific hours. The bar chart reveals that they are all of *low severity* and that they were all generated by applications of type ”remote control”. Figure 4 shows the details of the user/application view at this stage. The view clearly shows there is a cluster of users and applications which generate the majority of these alarms. In order to better discriminate between these resources, in Figure 7 we selected the reddest application which turned out to be an ”anonymous” application, that is an application whose name cannot be revealed for privacy reasons. As shown in the figure, we have been able to isolate a very precise group of users and one application that generate almost all the alarms of type ”scan and propagation”. Since the system permitted to isolate few users and their machines, it was possible to inspect the problem in closer details directly with them and to find a new rule to limit the usage of some specific applications which generated the majority of these kind of alarms.

6 Evaluation

The *SpiralView*’s design presented in this article is the result of various interactions with network security analysts from private companies who already use NEXThink’s engine for more than a year. Taking into account real world tasks, they provided us valuable feedbacks on the usefulness of our visualizations and on their usability. Other tools and features, not presented in this article, have been abandoned in the process. For instance, we developed a so called OriginalityView highlighting the most original applications and users, in respect to their usage of the network resources. The idea behind was to concentrate the analysis on the most interesting objects of the network. We also applied the RadViz technique as a first attempt to cluster similar users and applications. Although these tools received encouraging feedbacks from the analysts, more

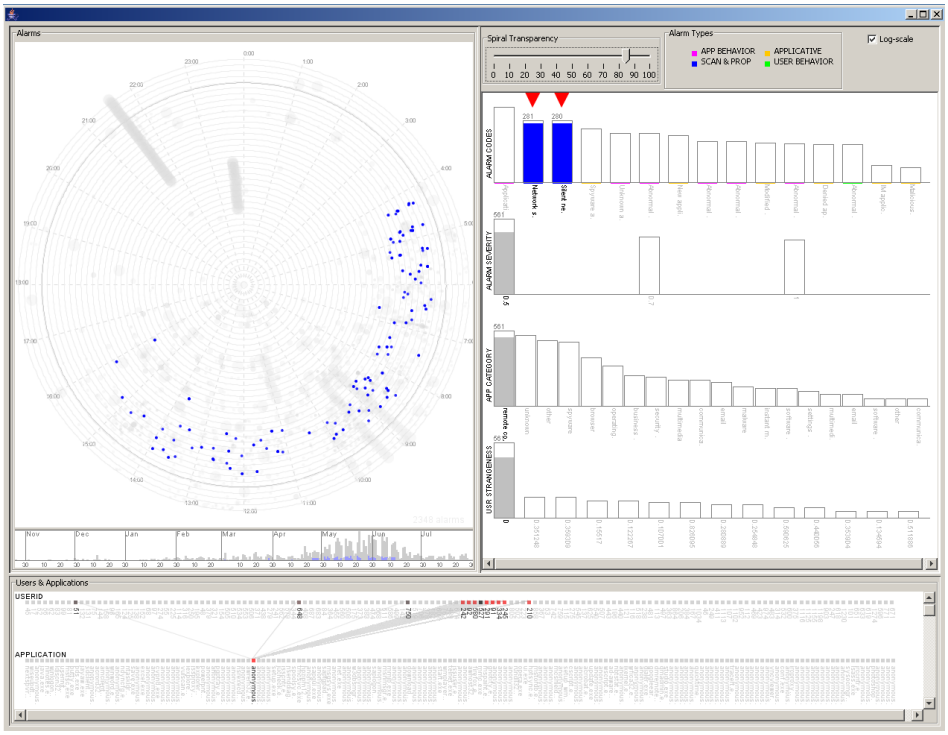

Figure 7: Case Study 2: the alarms of type ”scan and propagation” are investigated to understand how they distribute in the population and when they appear. The final image reveals that there is a specific cluster of users and one application generating almost all of them.

works remain to be performed in order to connect them with the *SpiralView*.

Concerning the design of the *SpiralView* itself, we firstly used the Parallel Coordinates technique to represent the overall alarms' footprint. Although this technique was first abandoned, we have recently integrated a refined version employed only to get detailed information on a selected group of alarms. Roughly put, finer grain modifications have been made to improve the interaction and readability of the actual *SpiralView* (zooming mechanisms, brush/link with histograms, etc.). In the near future, the *SpiralView* will be used in production by private companies, which will enable us, with log tracing mechanisms to gather quantitative measures on its real usage.

7 Future Work

One of the main goals of this research project is to visualize alarms in long period of time to elicit recurrent patterns, track abnormalities, inspect how they relate to users, applications, in order to assess the impact of a novel network policy. We already received positive feedbacks from the market concerning this visual artifact. *SpiralView* however needs further work to become both an overview visualization and an entry point to the whole network's data, so that it can be used not only as an overview of the network strange and suspicious activities, but as well as an entry point to any detailed information of the network (overview + zoom capabilities). In particular, we would like to concentrate on the following tasks:

- Understanding all users' behavior through the users who generate alarms, or how to use *SpiralView* as an entry point: Connect *SpiralView* with the rest of the network world to turn it to an entry point for the whole network's data. The spiral will then integrate overview and zoom, which are the two pillars of a successful interactive visualization. The idea is to permit the analyst to explore the resources involved in alarms and then, once few interesting cases have been spotted, use them to query other users, applications, etc., taken from the whole population to look for similar trends, correlations, and patterns.

- Tighter integration of mining and visualization techniques: Implement and integrate visualizations and established mining techniques. In particular, we are interested in clustering users and applications to detect interesting groups and to segment the user population. We believe that visualization and mining can extremely benefit each other: from one hand mining helps in reducing large data sets in fewer key cases and dimensions, from the other, visualization may help in representing mining results and explaining their relationship with the original data so that trust and understanding can be improved.

- Temporal visualization of network metrics such as the evolution of the number of alarms, types of alarms, criticality, etc. might be a good indicator of the health of a network and provide visual clues on action to perform. We have already started along these lines devising a new system able to analyze the temporal evolution of key metrics (to capture the risk level of a network) and to explain their variation.

8 Conclusions

We have presented in this article the *SpiralView*, a visualization tool for the analysis of security alarms in a corporate network. The tool permits to visualize alarms on an extended time span and facilitates long term analysis and strategies building. Differently to existing systems, it permits to abstract away from day-to-day monitoring and enables administrators and network managers to gain a big picture view of how security evolves over a long time period. The spiral visualization plots alarms in time, and coupled with interactive

bar charts is used to navigate through network data and perform queries. The user is able to segment alarms in meaningful subsets according to specific network resources and metrics, zoom on specific related information, and inspect for relationships between alarms, users, and applications. Future works include the use of *SpiralView* as an entry point to the all network's data in order to understand the behavior of the whole population of users through criminal users, and also the development of novel visualization of networks metrics in time, in order to predict and act on the network's evolution.

ACKNOWLEDGEMENTS

This work has been developed under the project "*Network Security Intelligence through Distributed User Behaviour Modelling and Interactive Visualizations*" funded by KTI/CTI (Swiss Confederation's innovation promotion agency)

REFERENCES

[1] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko. Ids rainstorm: Visualizing ids alarms. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC 2005)*, Minneapolis, MN, USA, October 2005.

[2] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM Press.

[3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[4] C. A. Brewer. Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60, Alexandria VA, 1999. American Statistical Association.

[5] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 29–38, New York, NY, USA, 1998. ACM Press.

[6] G. Conti, J. Grizzard, M. Ahamad, and H. Owen. Visual exploration of malicious network objects using semantic zoom, interactive encoding and dynamic queries. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC 2005)*, Minneapolis, MN, USA, October 2005.

[7] G. A. Fink and C. North. Root polar layout of internet address data for security administration. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 7, Washington, DC, USA, 2005. IEEE Computer Society.

[8] S. Foresti, J. Agutter, Y. Livnat, S. Moon, and R. Erbacher. Visual correlation of network alerts. *IEEE Computer Graphics and Applications*, 26(2):48–59, 2006.

[9] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Preserving the big picture: Visual network traffic analysis with tnv. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC 2005)*, Minneapolis, MN, USA, October 2005.

[10] C. G. Healey and J. T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[11] P. Hertzog. Visualizations to improve reactivity towards security incidents inside corporate networks. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC'06)*, Alexandria, VA, USA, November 2006.

[12] Y. Hideshima and H. Koike. Starmine : A visualization system for cyber attacks. In *Proceedings of Asia-Pacific Symposium on Information Visualization (APVIS 2006)*, Tokyo, Japan, February 2006.

[13] H. Koike and K. Ohno. Snortview: Visualization system of snort logs. In *Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC '04)*, Washington, DC, USA, October 2004.

[14] K. Lakkaraju, W. Yurcik, and A. J. Lee. Nvisionip: Netflow visualizations of system state for security situational awareness. In *Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC '04)*, Washington, DC, USA, October 2004.

[15] Q. Li and C. North. Empirical comparison of dynamic query sliders and brushing histograms. In *Proc. of the IEEE Symposium on Information Visualization 2003*, page 19, Los Alamitos, CA, USA, 2003.

[16] K. Mundiandy. Case study: Visualizing time related events for intrusion detection. In *Proc. of the IEEE Symposium on Information Visualization 2001*, pages 22–23, Washington, DC, USA, 2001. IEEE Computer Society.

[17] R. Spence and L. Tweedie. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11:137–146, 1998.

[18] N. Stanton, editor. *Human factors in alarm design*. Taylor & Francis, Inc., Bristol, PA, USA, 1994.

[19] T. Takada and H. Koike. Mielog: A highly interactive visual log browser using information visualization and statistical analysis. In *Proceedings of the Sixteenth Systems Administration Conference (LISA '02)*, Berkeley, CA, USA, November 2002.

[20] T. Takada and H. Koike. Tudumi: Information visualization system for monitoring and auditing computer logs. In *Proceedings of the Sixth International Conference on Information Visualisation (IV'02)*, London, England, UK, July 2002.

[21] L. B. Ward M. A visualization tool for exploratory analysis of cyclic multivariate data. *Metrika*, 51:27–37, 2000.

[22] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[23] M. Weber, M. Alexa, and W. Muller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 7, Washington, DC, USA, 2001. IEEE Computer Society.

[24] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proc. of the IEEE Symposium on Information Visualization 2001*, page 43, Washington, DC, USA, 2001. IEEE Computer Society.

[25] W. Yurcik. Visflowconnect-ip: A link-based visualization of netflows for security monitoring. In *Proceedings of the Eighteenth Annual FIRST Conference on Computer Security Incident Handling*, Baltimore, MD, USA, June 2006.