Visual Filter: Graphical Exploration of Network Security Log Files

Jan-Erik Stange
FH Potsdam
Kiepenheuerallee 5
stange@fh-potsdam.de

Marian Dörk
FH Potsdam
Kiepenheuerallee 5
doerk@fh-potsdam.de

Johannes Landstorfer
FH Potsdam
Kiepenheuerallee 5
landstorfer@fh-potsdam.de

Reto Wettach
FH Potsdam
Kiepenheuerallee 5
wettach@fh-potsdam.de

ABSTRACT

Network log files often need to be investigated manually for suspicious activity. The huge amount of log lines complicates maintaining an overview, navigation and quick pattern identification. We propose a system that uses an interactive visualization, a visual filter, representing the whole log in an overview, allowing to navigate and make context-preserving subselections with the visualization and in this way reducing the time and effort for security experts needed to identify patterns in the log file. This explorative interactive visualization is combined with focused querying to search for known suspicious terms that are then highlighted in the visualization and the log file itself.

Categories and Subject Descriptors

D.2.2 [**SOFTWARE ENGINEERING**]: Design Tools and Technique – *user interfaces, evolutionary prototyping*

General Terms

Design, Experimentation, Security

Keywords

Human Pattern Recognition, Visual Filter, Dynamic Querying, Exploratory Search, Overview and Detail

1. INTRODUCTION

Protecting large computer networks from security risks is a process that involves huge amounts of data to be handled. Usually this is done with algorithms that automatically detect known threats and inform security experts with aggregated events. This automatic analysis is a necessary step in security monitoring since it has become impossible to manually analyze the security-relevant data in real-time.

Although this automatic detection unburdens the security expert monitoring the network considerably, it is far from perfect and depends on the security expert's continuing effort in defining appropriate rules for detection. Network owners have to rely on intrusion detection systems and deal with the many false positives they produce to be able to enforce network security.

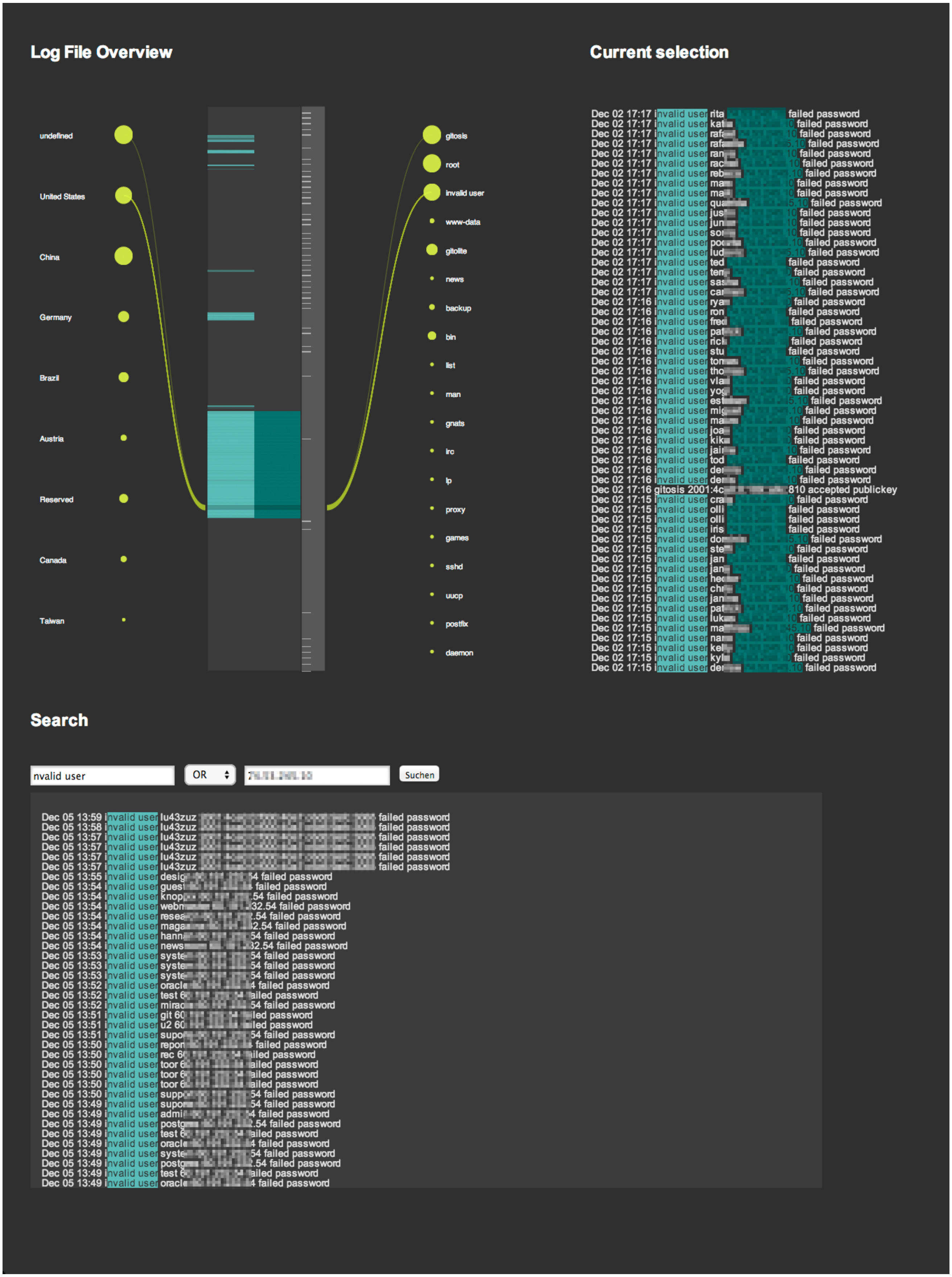
There are often cases when suspicious activity in a network goes unnoticed by the intrusion detection systems because no rules have so far been defined to identify them. New threats are often found (if they are found at all) at a later point in time by manually analyzing typical network data like server log files or netflows [2]. This analysis after an incident has occurred is referred to as incident analysis. The insight gained from this analysis is then often used to define new rules for the intrusion detection system.

When looking at the raw data of single logs or netflows the analyst is confronted with considerably large amounts of textual data that they have to process and filter. This is usually done by applying command-line tools like GREP or PERL scripts with regular expressions. Although fast and transparent, this process has several drawbacks. For one thing it is hard to gain an overview over such a file by only looking at and working with the raw text. Another issue is the difficult navigation within these files with thousands of lines. A way around this is filtering down the content to a relevant subset. The problem with this is, however, that it is not always straightforward to define what is relevant. Suspicious activity usually manifests itself in certain patterns, combinations of particular kinds of log lines, that are hard to distinguish from the surrounding log lines. Marty gives a detailed account of the ways in which visualization offers benefits over textual analysis [10].

In order to overcome the above-mentioned issues of command-line analysis we developed a set of principles we subsume under the concept visual filter. We will present and elaborate on these principles in section 3.

In order to validate our concept we applied it to a system called LogSpider that uses a visualization that serves as a visual

VizSec '14, November 10 2014, Paris, France Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-2826-5/14/11…$15.00 http://dx.doi.org/10.1145/2671491.2671503



Log File Overview

Current selection

```
Dec 02 17:17 invalid user rita [redacted] failed password
Dec 02 17:17 invalid user kat[redacted]0 failed password
Dec 02 17:17 invalid user rafa[redacted]10 failed password
Dec 02 17:17 invalid user rafa[redacted]5.10 failed password
Dec 02 17:17 invalid user ran[redacted]10 failed password
Dec 02 17:17 invalid user rac[redacted]10 failed password
Dec 02 17:17 invalid user reb[redacted].10 failed password
Dec 02 17:17 invalid user ma[redacted]0 failed password
Dec 02 17:17 invalid user ma[redacted]0 failed password
Dec 02 17:17 invalid user qua[redacted]5.10 failed password
Dec 02 17:17 invalid user jus[redacted]10 failed password
Dec 02 17:17 invalid user jun[redacted]10 failed password
Dec 02 17:17 invalid user sor[redacted]10 failed password
Dec 02 17:17 invalid user poc[redacted].10 failed password
Dec 02 17:17 invalid user lud[redacted]5.10 failed password
Dec 02 17:17 invalid user ted [redacted] failed password
Dec 02 17:17 invalid user ten[redacted]0 failed password
Dec 02 17:17 invalid user sas[redacted]10 failed password
Dec 02 17:17 invalid user car[redacted]5.10 failed password
Dec 02 17:16 invalid user rya[redacted]0 failed password
Dec 02 17:16 invalid user ron [redacted] failed password
Dec 02 17:16 invalid user fred [redacted] failed password
Dec 02 17:16 invalid user pat[redacted].10 failed password
Dec 02 17:16 invalid user rick [redacted] failed password
Dec 02 17:16 invalid user stu [redacted] failed password
Dec 02 17:16 invalid user tom[redacted]10 failed password
Dec 02 17:16 invalid user tho[redacted]5.10 failed password
Dec 02 17:16 invalid user vla[redacted]0 failed password
Dec 02 17:16 invalid user yog[redacted]0 failed password
Dec 02 17:16 invalid user est[redacted]5.10 failed password
Dec 02 17:16 invalid user mig[redacted].10 failed password
Dec 02 17:16 invalid user ma[redacted]10 failed password
Dec 02 17:16 invalid user joa[redacted]0 failed password
Dec 02 17:16 invalid user kik[redacted]0 failed password
Dec 02 17:16 invalid user jair[redacted]10 failed password
Dec 02 17:16 invalid user tod [redacted] failed password
Dec 02 17:16 invalid user der[redacted].10 failed password
Dec 02 17:16 invalid user den[redacted]10 failed password
Dec 02 17:16 gitosis 2001:4c[redacted]810 accepted publickey
Dec 02 17:15 invalid user cra[redacted]0 failed password
Dec 02 17:15 invalid user olli [redacted] failed password
Dec 02 17:15 invalid user olli [redacted] failed password
Dec 02 17:15 invalid user iris [redacted] failed password
Dec 02 17:15 invalid user dom[redacted]5.10 failed password
Dec 02 17:15 invalid user ste[redacted]0 failed password
Dec 02 17:15 invalid user jan [redacted] failed password
Dec 02 17:15 invalid user jan[redacted]0 failed password
Dec 02 17:15 invalid user hec[redacted]10 failed password
Dec 02 17:15 invalid user chr[redacted]0 failed password
Dec 02 17:15 invalid user jan[redacted]10 failed password
Dec 02 17:15 invalid user pat[redacted].10 failed password
Dec 02 17:15 invalid user luk[redacted]10 failed password
Dec 02 17:15 invalid user ma[redacted]45.10 failed password
Dec 02 17:15 invalid user nar[redacted]0 failed password
Dec 02 17:15 invalid user kel[redacted]0 failed password
Dec 02 17:15 invalid user kyl[redacted]0 failed password
Dec 02 17:15 invalid user der[redacted].10 failed password
```

Search

nvalid user | OR | [redacted] | Suchen

```
Dec 05 13:59 invalid user lu43zuz [redacted] failed password
Dec 05 13:58 invalid user lu43zuz [redacted] failed password
Dec 05 13:57 invalid user lu43zuz [redacted] failed password
Dec 05 13:57 invalid user lu43zuz [redacted] failed password
Dec 05 13:57 invalid user lu43zuz [redacted] failed password
Dec 05 13:57 invalid user lu43zuz [redacted] failed password
Dec 05 13:55 invalid user desig[redacted]54 failed password
Dec 05 13:54 invalid user guest[redacted] failed password
Dec 05 13:54 invalid user knop[redacted].54 failed password
Dec 05 13:54 invalid user webm[redacted]32.54 failed password
Dec 05 13:54 invalid user resea[redacted].54 failed password
Dec 05 13:54 invalid user maga[redacted]2.54 failed password
Dec 05 13:54 invalid user hann[redacted]54 failed password
Dec 05 13:54 invalid user news[redacted]32.54 failed password
Dec 05 13:53 invalid user syste[redacted]54 failed password
Dec 05 13:53 invalid user syste[redacted]54 failed password
Dec 05 13:53 invalid user syste[redacted]54 failed password
Dec 05 13:52 invalid user oracle[redacted]4 failed password
Dec 05 13:52 invalid user test 6[redacted]ailed password
Dec 05 13:52 invalid user mirac[redacted]54 failed password
Dec 05 13:51 invalid user git 60[redacted]iled password
Dec 05 13:51 invalid user u2 60[redacted]iled password
Dec 05 13:51 invalid user supo[redacted]54 failed password
Dec 05 13:50 invalid user repor[redacted] failed password
Dec 05 13:50 invalid user rec 60[redacted]iled password
Dec 05 13:50 invalid user toor 6[redacted]ailed password
Dec 05 13:50 invalid user toor 6[redacted]ailed password
Dec 05 13:50 invalid user toor 6[redacted]ailed password
Dec 05 13:50 invalid user supp[redacted]54 failed password
Dec 05 13:49 invalid user supor[redacted]54 failed password
Dec 05 13:49 invalid user admi[redacted]4 failed password
Dec 05 13:49 invalid user postg[redacted]2.54 failed password
Dec 05 13:49 invalid user test 6[redacted]ailed password
Dec 05 13:49 invalid user oracle[redacted]4 failed password
Dec 05 13:49 invalid user syste[redacted]54 failed password
Dec 05 13:49 invalid user postg[redacted]2.54 failed password
Dec 05 13:49 invalid user test 6[redacted]ailed password
Dec 05 13:49 invalid user oracle[redacted]4 failed password
```

Figure 1: Overview over the components Graphical Overview, Current Selection and Query Search

filter that gives an overview, allows fast navigation in the log file and selection of subsections that are displayed without losing the overall context. The interactive graphical overview representation of the log file is accompanied by a query interface that allows to specify regular expressions. The results are then highlighted in the overview representation creating visual patterns within the surrounding context. The intention behind the combination of an explorative graphical browsing interface and a focused search interface is to support exploratory search [9] in the context of network security log analysis.

In the following section our approach will be put in relation to existing research. In Section 3 we will elaborate on our design concept and describe the different parts of the interface. We evaluated our visualization with three security experts. The use case and data we focused on and the results of the evaluation will be discussed in Section 4: Evaluation. The paper closes with Section 5: Conclusion and Future Work.

2. RELATED WORK

There are several tools for visualizing the contents of computer log files. These can be categorized in terms of what data they use for visualization. One common method is applying statistical methods to create aggregations which are then used to build a visual overview over the whole file. The Histogram Matrix [4] uses a matrix visualization in which the frequency of log message length is plotted against time. The higher the frequency of a certain message length, the bigger the circle gets at the according position. These circles can then be selected and all the messages with that length are shown. ELVIS [6] offers a view of small thumbnail histograms of the different data dimensions in a log file as a starting point for further investigation. These thumbnails can be dragged to an area where they are combined to new visualizations like heatmaps, pie charts etc. The algorithm used for the LogView [8] visualization exceeds simple statistical calculations and identifies particular clusters of log messages as well as outliers and renders them in an interactive treemap layout.

Another approach is to incorporate the textual structure of the log file in the visualization. Apart from some statistical aggregations to navigate the log, MieLog [12] includes an “outline area” that shows the length of single log messages with pixel lines in their consecutive order in the file. The SDSS Log Viewer [14] goes a step further by color-coding the fragments of user-generated SQL database queries combining them to pixel lines with different colors. A slightly dated approach is described in SeeSoft [2]. Despite its datedness and its application to a non-security-specific context it is relevant for our research since it features visual concepts that are similar to the ones we applied. Log lines generated by a software to run telecommunications switches are rendered as single pixel lines that are color-coded by message type. This overview can be enhanced by “browsing windows” that allow users to select parts of the visual overview with little rectangles and in this way generate windows that display the text represented by the pixel lines in detail.

In contrast to the Histogram Matrix, ELVIS and LogView, MieLog, SDSS Log Viewer and SeeSoft keep the original structure of the log messages embedded in their context, which we argue is an important support for finding especially yet unknown patterns. MieLog offers a juxtaposition of graphical representation and textual content, with the limitation that only a fragment of the log is visually represented. SeeSoft displays a whole log file in an overview on one screen, but is limited in terms of the overall number of log lines by screen size and resolution since it doesn’t use any kind of aggregation. Our system provides a visual overview in a defined area of the screen over the whole log at all times independent of its size ensuring a strong sense of orientation. This is achieved by something that we call “visual aggregation”, meaning that one pixel line of our overview representation represents a certain amount of log lines depending on the overall log file size. When a search term has been entered, these pixel lines are highlighted with varying brightness depending on the frequency of the term in a particular sequence of lines represented by the pixel line. A similar approach has been documented in TileBars [5]. In this search user interface full-text documents are split into topical sections, which are then represented by small tiles with differing gray values depending on the frequency of specified search terms in that section. In contrast to our approach, these tiles only serve as an overview and cannot be selected to get a detail view. SeeSoft offers a textual detail view (browsing windows), but it is quite limited in the amount of information it can display since it offers only one detail level. LogSpider lets users progressively adjust the level of detail. Another drawback of these browsing windows is their overlap with the overview visualization and the lack of a designated area for them. Lastly, we emphasize the importance of supporting the user with an exploratory search process. While MieLog has predefined keywords implemented into the system that are highlighted in the text, it does not offer an advanced and flexible focused search with regular expressions, a necessary premise for exploratory search. LogView offers only a very basic query search that is not accompanied by an advanced browsing interface, since interaction with the treemap is limited to very basic operations.

3. LOGSPIDER

Common techniques of manual log file processing in incident analysis are text-based and require boiling down the log file to a manageable amount of lines. This only is applicable though, if the security analyst knows what they have to look for. If that is not the case, they run into the risk of losing the overview and filtering out valuable patterns hinting to new forms of threats. So for cases of yet unknown threats another approach is needed.

3.1 Towards Visual Filters in Log Analysis

Since information visualization is a powerful approach to amplifying human cognitive capabilities [11], we reasoned that we could make use of the benefits of visualization to develop a tool that especially focuses on finding visual solutions for the afore-mentioned problems of manual textual log file analysis. Specifically, with visual means we strive to come up with a visual analytics application that addresses the shortcomings in manual textual analysis with four interrelated principles:

- Graphical overview at all times
- Fast navigation without losing orientation
- Selecting subsets while keeping the context
- Exploratory search instead of sequential querying

These guidelines and their practical application to LogSpider will be detailed out in the following sections and in this combination be referred to as visual filter. Exploratory search is defined as a behavior that can be observed in cases when searchers are either [9]: (1) unfamiliar with the domain of their goal, (2) unsure about the ways to achieve their goals or (3) even unsure about their goals in the first place.

While the first one will usually not be an issue, the second and third are naturally common when a new, yet unknown security threat has manifested within a network. Addressing this kind of search behavior our proposed tool is well-equipped to meet the needs of an incident analysis process. In practice, exploratory search can be achieved by offering the user both browsing and query search options combined in one interface. We will discuss our strategy for achieving this in the following section.

When investigating unknown threats, the analyst offered exploratory search features can then start with browsing the log in a first step to identify potential areas for analysis (time spans or connections from particular countries, for example). In this way we hope to enable them to conduct an informed reduction of the data avoiding to filter out essential parts for the analysis from the beginning because of an insufficient understanding of the threat. This subselection can then be further reduced by using query search to highlight certain terms and extend the analysis of this subselection. The analyst might learn something new by investigating this part of the data that could lead them to the impression that they have to readjust their initial selection. We address this by allowing them to easily go back to the graphical overview and change the initial selection offering a flexible adaptation to a presumable ever increasing understanding of a threat with proceeding exploration. With our concept of a visual filter it is our intention to provide for a continuous interplay between browsing and query search, a necessary premise for exploratory search, as defined in [9], leading to an informed reduction of the log data to be analyzed.

Aside from the particular research on incident analysis in security log files, we conducted several interviews and a workshop in the larger context of our research project on network security monitoring with security analysts, network monitoring specialists and scientific researchers in the field of security that gave us a general impression of important goals and user needs in that area (see [7] for a detailed account of the co-creation approach). Based on these insights, five key design goals emerged for visual analytics tools in network security in our project:

1 Display raw data

Because of the high rate of false positives network analysts want to check on the data that algorithms used to produce an alert. Raw data should be made readily available to apply human reasoning when in doubt.

2 Visualize extrema

Often either rare events or events with high frequency are particularly suspicious. Visualizations should take care to especially emphasize these kinds of events. While frequent events naturally catch the eye by their amount, rare events should be highlighted to make sure they are not missed by the analyst.

3 Encourage explorative analysis

As described further up, in cases of uncertainty of the goal of a search or the way to achieve it, the user should be provided with an exploratory search interface.

4 Combine overview and quick filtering

Visual tools should help experts to stay in context, correlate events and find patterns. While the overview can point you to larger patterns spreading over substantial amounts of the data, certain threats can only be identified with a detail view of the data. Still, it is important to see that detail in the context of the surrounding data and the dataset as a whole to be able to make correlations.

5 Support various log file types

A general problem in network management and network security is the abundance of different log data types, formats by different manufacturers and the different configurations you can choose regarding what will be written into the log file. Not having to write your own parser for each new kind of dataset, would streamline the incident analysis process.

These goals served as a more general framing for our design and development process and helped establish the above-mentioned four principles for the design of the LogSpider.

3.2 Interface Components

The user interface of LogSpider consists of three components: The Graphical Overview, the Current Selection and the Query Search (see Figure 1). Since they all operate on the same data basis, the log file, they naturally are connected with each other: Manipulating one component usually influences the state of the other components. Selecting a section of the Graphical Overview automatically changes the view in the Current Selection accordingly, while starting a search with a regular expression in the Query Search component highlights the matching parts in the Graphical Overview as well as in the Current Selection, in addition to the actual displayed search results in the Query Search component. This method is known as brushing and linking [1].

The two components Graphical Overview and Current Selection are designed to support an exploratory analysis of the log. When a first exploration provides a better idea of what to look for exactly, the Query Search allows the user to specify more concrete queries. The investigation of parts of the log with highlighted query terms in the Graphical Overview and the Current Selection are intended to support a more thorough understanding of the problem and might lead to another query with regular expressions. This interplay between exploratory browsing strategies and focused search can then be continued until the problem has been encircled.

3.3 The Graphical Overview

The Graphical Overview provides an overview over the whole log file with a number of juxtaposed dark gray pixel lines representing a certain amount of log lines depending on the total amount of lines in the file. These “visual bins” are highlighted, when matching lines are found within them. The higher the number of matches per line, the higher the brightness of the respective pixel line. The pixel lines are split, so that the results of two search queries can be shown in juxtaposition with two different colors. The search queries are connected with boolean operators. On the right side of the gray representation there is a slightly brighter area with small marks. The area between two marks represents a certain amount of time that is set to two hours. In future iterations users should be able to adjust the time span. The marks can be hovered over to show the point of time as a label. This part of the Graphical Overview is designed to help identifying periods in the log file with an unusually high amount of entries, which can often be a first hint that some security issue is present. A fragment of the entire log can be selected by clicking and dragging in the pixel-lines area, creating a brush, which can then be moved and resized. The fragment that is brushed at any given time is then shown on the right-hand side in plain text as the Current Selection.

On both sides of the visual bins light green circles can be seen that visualize the frequency of values of a particular data dimension of the log by their size. In Figure 1 the two dimensions country (derived from the source IP address of the log file) and username are chosen as an example (In future versions users should be able to choose appropriate dimensions for their particular use cases). In addition to serving as a histogram, these circles also function as a special kind of network nodes that are connected with curved lines to the brush area, if the lines in the brush area contain the value of a particular node. The nodes can also be clicked to show all occurrences of one value (Clicking the nodes provides a visual hint to occurrences, it does not serve as a data filter changing the contents of the Current Selection or the Query Search, since this would lead to context-loss, something we were careful to avoid).

3.4 The Current Selection

This area shows the log lines that correspond to the chosen brush in the Graphical Overview. If the user has already started a search in the Query Search, found matches are highlighted in the Current Selection as well. The size of the displayed text in this component is dependent on the size of the brush in the Graphical Overview and changes dynamically when resizing the brush: The bigger the brushed area, the smaller the displayed log lines get, to the point, where the user receives more of a general impression of the structure of the selected part of the log. Unusual lengths of log lines stand out visually and in effect hint to unusual events, while highlighted words are still visible. With iterative adjustment of the brush suspicious areas can be quickly put into focus again.

3.5 The Query Search

The Query Search offers two query fields for specifying queries that can be regular expressions or plain text entries. These two fields can be connected with a boolean operator AND or OR. Plain text search can be helpful in cases, when a user has found a term in the Current Selection, for which they want to find all occurrences in the log file. They can then quickly copy and paste that term and start the search. For a more sophisticated search that is comparable to the powerful possibilities of command-line-analysis with GREP, for example, they can specify complex regular expressions. Each field has a certain color assigned that is used in the Graphical Overview and Current Selection accordingly. The results of a search, e.g. all the matching lines, are displayed in a box below the query fields.

3.6 Example Walkthrough

To illustrate the mechanics of the system we describe a typical log analysis scenario, in which a pre-processed SSH server log file has been loaded into LogSpider. In Figure 2 we can see a screenshot of LogSpider. As is apparent, in the lower part of the Graphical Overview (the newest entry is at the top) there are several ticks with wider gaps between them, meaning there was a larger amount of requests to the server in these two-hour spans. This can already be a first indicator for suspicious activity. About half of one of these larger areas has been selected and on the left side the analyst can observe almost all lines going from the selected brush to either a node with the label "undefined" or "China". Since they know that in their network only IPv6-addresses are assigned and they also know that the big number of "undefined" locations arises from the inability of the geo-ip-service they chose to resolve these IPv6-addresses, the analyst can assume that these are probably their own requests within the network. The huge amount of requests from a country they knew they did not have employees in at that time, is something that arouses suspicion, on the other hand. Also, the analyst notices an unusual number of lines going to "root" on the right side. In the Current Selection on the right side they can also see the monotony of the requests, only from time to time interrupted by longer lines. In Figure 3 the size of the brush has been reduced, this way zooming into the detail view of the Current Selection. In contrast to the state before, where the log lines formed more like a general pattern without revealing its contents, the analyst can now read the contents of single lines and recognizes there are a lot of lines with the same IP address and a log message "failed password", a clear sign that there was a brute-force attack in progress. In between these sequences, connection attempts by a user of the analyst's network can be observed, which have a longer line length because of an IPv6 address and a different log message ("accepted publickey"). If the analyst wants to investigate the

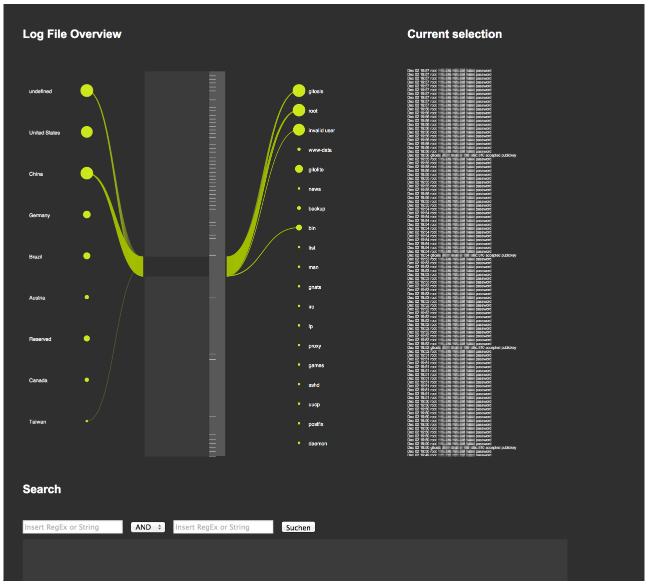
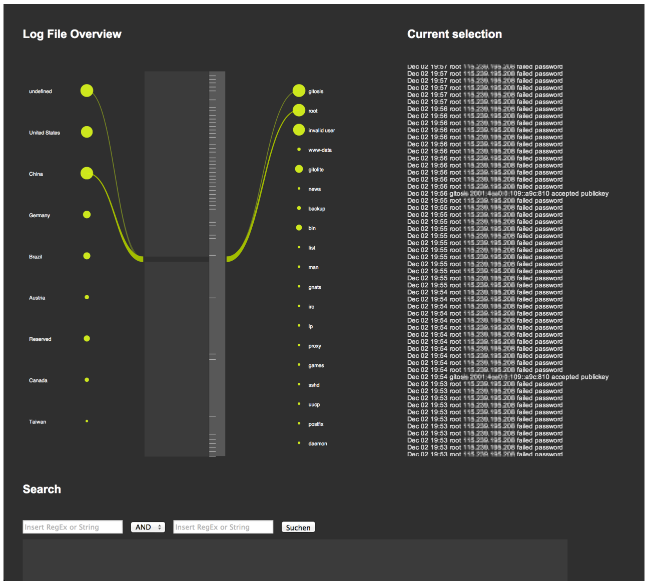
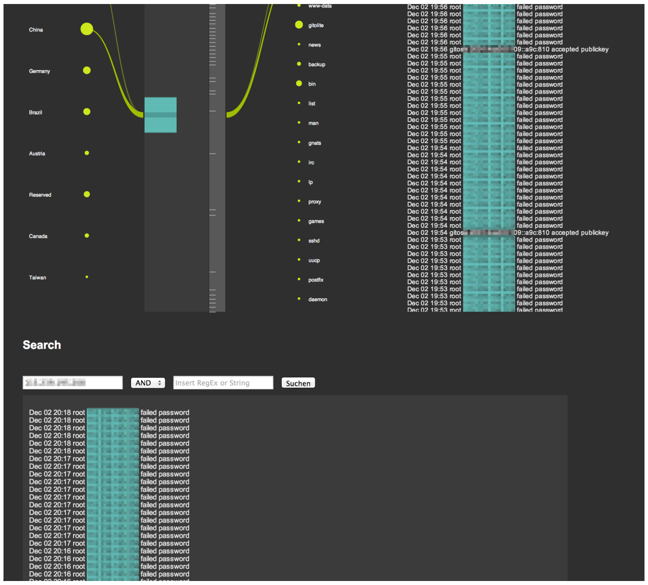


Figure 2: Selection of area with high amount of traffic



```
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
```

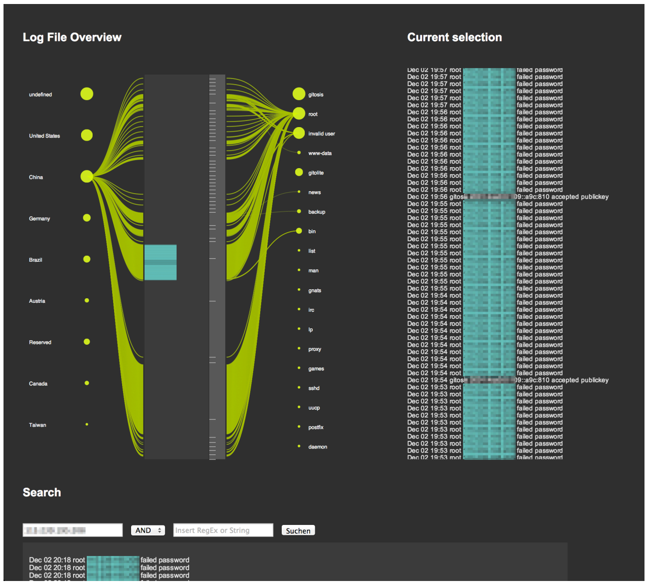
Figure 3: Zooming into area



```
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
```

```
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:17 root [redacted] failed password
Dec 02 20:16 root [redacted] failed password
Dec 02 20:16 root [redacted] failed password
Dec 02 20:16 root [redacted] failed password
Dec 02 20:16 root [redacted] failed password
```

Figure 4: Query search started with suspicious IP



```
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:57 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 root [redacted] failed password
Dec 02 19:56 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:55 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 root [redacted] failed password
Dec 02 19:54 gitos[redacted]09::a9c:810 accepted publickey
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
Dec 02 19:53 root [redacted] failed password
```

```
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
Dec 02 20:18 root [redacted] failed password
```

Figure 5: All occurrences of one IP

potentially malicious IP address in more detail, they can make use of the Query Search component. In Figure 4 we can see the result of a simple search for the above-mentioned IP address. The analyst can just copy the address from the Current Selection and paste it into the input field. The IP address is now highlighted everywhere it occurs in the log file in the Graphical Overview as well as in the Current Selection. Naturally, the most important question the analyst wants to answer is, if an attack was successful and measures for mitigation have to be taken. Usually a brute-force attack stops, when a correct password has been guessed, so, if the attack had been successful, the last log line in the search results should be "accepted password". As we can see, the last (topmost) entry in the search results from Dec 02, 20:18 is a "failed password", so the analyst can be relatively sure that this attack was not successful. As mentioned before, the size of the particular country node on the left is considerably big, so the analyst wants to check at what times in the log file connection attempts from the respective country were documented. By clicking on the node, they can visualize all occurrences of IP addresses from that country in the log file. As can be seen in Figure 5, there are several blocks of increased activity of Chinese IP addresses. The analyst can now continue exploring these other blocks in a similar manner.

4. EVALUATION

To validate our approach and study the potential of visual filters for log analysis, we were following an iterative design approach that included internal reviews and walkthroughs of our interactive prototype within our research group as well as receiving feedback from two security experts from the Leibniz Supercomputing Centre (LRZ) in Munich and a security expert from a security consultancy in Frankfurt/Main. We were specifically concentrating on the use case "brute-force attack".

4.1 Prototypical Implementation

The user interface of LogSpider is implemented as a web application using current technologies and standards (HTML5, CSS3 and Javascript, as well as JavaScript libraries jQuery and D3).

To put our hypothesis that visual filters simplified pattern recognition in log files to the test, we had to work with real-world data. One of our partners in the research project we are engaged in, the LRZ, provided us with two SSH server log files of one of the servers in their network. This datasets satisfied the requirements that we had to look for to generate meaningful results: They were too large to exclude the possibility of an analysis by scrolling through it (> 3000 lines), they were generated in a large network that is exposed to persistent attacks and they contained real attacks from different origins. The datasets were pre-processed by our partner at the LRZ to exclude log lines that can be neglected when looking for security threats and exported as text files in csv-format. Each of the datasets consisted of the data dimensions time, destination IP, destination port, log message, source IP, source port, user name and protocol.

4.2 Expert Feedback

The experts we consulted evaluated the value of LogSpider with regard to the efficiency in finding these attacks compared to other more text-based approaches they apply (GREP and other regular-expression-based tools), but also with regard to more general problems in finding unusual patterns in log files.

Though the system couldn't play out its full potential with this particular kind of dataset, because the data contained only three kinds of log messages (accepted publickey, accepted password, failed login) and because of that only a limited variation in the texts to be analyzed, it performed well and all attacks could be found. We have to mention here that, since the threat patterns in the log file were of rather low complexity and well-known, it would have been relatively easy to find them with the above-mentioned conventional text-based approaches as well.

Nonetheless, the result of this expert evaluation was encouraging and an application of the visualization to log files with more diverse log messages seems promising for our tool, especially when the use case is more complex than a brute force attack, which leaves a simple pattern in the log, as one of the experts emphasized.

The usefulness of having both the explorative browsing side-by-side with the focused search was stressed by one of the experts. At an earlier stage in the design, we provided only the explorative part.

Another point of criticism were the nodes in the Graphical Overview. With respect to the number of different values in one column of the dataset (sometimes up to thousands as was the case for "username"), it seemed impossible to display them all as nodes and still keep an overview. The experts suggested to aggregate these values. A security expert doesn't need to know the names of all the invalid users, so these could be aggregated. Also, it seemed more reasonable to resolve the IP addresses to countries and display these countries as nodes. These suggestions have already been implemented in the current version of the tool.

There was a proposition to make better use of the space used for the Graphical Overview by visualizing the sequences of different log messages with single pixels with varying colors for each log line.

Since threats in networks often manifest in more than one network component and thus in several log files, one expert suggested to include more than one log file in the visualization. This could allow network analysts to investigate orchestrated attacks across the network.

5. CONCLUSION AND FUTURE WORK

In this paper we presented LogSpider, a visualization system for network log files that was specifically designed to support a faster and more effective discovery of security issues through exploratory log analysis by applying a concept called visual filter that has not been investigated in security visualization until now.

Our interactive visualization is tightly bound to the dataset to be analyzed and offers an overview and a sense of orientation. It has been designed to allow quick navigation and selection of relevant subsets of the file without losing context. These features have been found useful by the experts that were involved in our iterative design process. Nonetheless, these preliminary findings have to be validated with more complex datasets, especially when several log files are analyzed in parallel, as was suggested by the experts. Part of the concept of the visual filter was to provide an interface that supports explorative browsing as well as focused search, to account specifically for use cases in which the goal of the search is not yet clear to the security expert or it is not yet clear how they could achieve their goal. In our evaluation this was considered a helpful approach by the experts even with a rather simple form of pre-processed log we received from our partners. Our assumption is that a more complex log file with more diverse content would be better-suited. In a further iteration we will look for other datasets that might allow more accurate inferences on how the LogSpider performs. A dataset with yet unknown security issues for the security expert would be ideal. To test this it would be best to conduct a field study and hand the LogSpider over to security experts to let them test it in their own network environment, since we obviously cannot ask security experts to provide us with network log files that contain threats they do not know about yet.

So far, the feedback we received is encouraging and following up on our system seems to be promising. In addition to the above-mentioned need for new test settings with more diverse and complex datasets that ideally contain unknown threats, we would like to offer the user the option to display different logs in parallel in the next iteration. We will also rethink our approach to representing the log lines. One suggestion was that single lines could be displayed as single pixels, in this way maybe even rendering aggregation of the log lines redundant. It would presumably be even more helpful to detect certain patterns, if there was a way to switch to a non-aggregated visualization.

ACKNOWLEDGMENTS

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under support code 01BP12303; EUREKA-Project SASER.

Special thanks to Leibniz Super Computing Centre for helping with test data sets and intense discussion and support.

6. REFERENCES

[1] Becker, R. A., Cleveland, W. S. 1987. *Brushing Scatterplots.* In Technometrics 29, 2 (May 1987), 127-142. DOI= http://dx.doi.org/10.2307/1269768

[2] D'Amico, A. D., Goodall, J. R., Resone, D. R., Kopylec, J. K. 2007. Visual Discovery in Computer Network Defense. In: Computer Graphics and Applications, IEEE. Volume 27 Issue 5 (Sept.–Oct. 2007). 20 – 27

[3] Eick, S. G., Steffen, J. L., Sumner Jr., E. E. 1992. *Seesoft: A Tool for Visualizing Line Oriented Software Statistics.* In IEEE Transactions on Software Engineering – Special issue on software measurement principles, techniques and environments. Volume 18 Issue 11 (November 1992), 957–968.

[4] Frei, A., Rennhard, M. 2008. *Histogram Matrix: Log File Visualization for Anomaly Detection.* In Proceeding of: Availability, Reliability and Security. (ARES 08). *Third International Conference on.* Barcelona, Spain, March 4-7, 610-617. DOI=10.1109/ARES.2008.148

[5] Hearst, M. 1995. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '95. Denver, CO, 59 – 66.

[6] Humphries, C., Prigent, N., Bidan, C., Majorczyk, F. 2013. *ELVIS: Extensible Log Visualization.* In Proceedings of the Tenth Workshop on Visualization for Cyber Security. VizSec '13. Atlanta, GA, USA, October 14, 2013, 9-16.

[7] Landstorfer, J., Herrmann, I., Stange, J.-E., Dörk, M., Wettach, R. 2014. Weaving a Carpet from Log Entries: A Network Security Visualization Built with Co-creation. IEEE Conference on Visual Analytics Science and Technology 2014. To appear.

[8] Makanju, A., Brooks, S., Zincir-Heywood, A. N., Milios, E. E. 2008. *LogView: Visualizing Event Log Clusters.* In: Sixth Annual Conference on Privacy, Security and Trust. PST '08. Fredericton, NB, October 1–3, 2008, 99 – 108.

[9] Marchionini, G. 2006. Exploratory Search: From Finding to Understanding. In Communications of the ACM – Supporting Exploratory Search Vol. 49, 4 (April 2006), 41-46. DOI= http://dl.acm.org/citation.cfm?doid=1121949.1121979.

[10] Marty, Raffael. 2009. Applied Security Visualization. Boston, MA, USA: Pearson Education.

[11] Card, S., Mackinlay, J.D., Shneiderman, B. 1999. *Readings in Information Visualization: Using Vision to Think.* San Francisco, CA: Morgan Kaufmann Publishers.

[12] Takada, T., Koike, H. 2002. *Mielog: A Highly Interactive Visual Log Browser Using Information Visualization and Statistical Analysis*. In Proceedings of the 16th USENIX Conference on System Administration. LISA '02. Philadelphia, PA, November 3-8, 2002, 133 – 144.

[13] White, Ryen W, Roth Resa A. 2009. *Exploratory Search: Beyond the Query-Response Paradigm*. San Rafael, CA: Morgan and Claypool. DOI= 10.2200/S00174ED1V01Y200901ICR003.

[14] Zhang, J., Chen, C., Vogeley, M., Pan, D., Thakar, A., Raddic, J. 2012. *SDSS Log Viewer: Visual Exploratory Analysis of Large-Volume SQL Log Data*. In Proceeding of Visualization and Data Analysis (VDA). Burlingame, CA, USA. January 22, 2012.