Monitoring Large IP Spaces with ClockView

Christopher Kintzel
University of Konstanz
Germany
Christopher.Kintzel@uni-konstanz.de

Johannes Fuchs
University of Konstanz
Germany
Johannes.Fuchs@uni-konstanz.de

Florian Mansmann
University of Konstanz
Germany
Florian.Mansmann@uni-konstanz.de

ABSTRACT

The growing amounts of hosts that are placed into the networks represent an enormous challenge to most network administrators who have to monitor these hosts conscientiously. While automatically monitoring the network for slow or failing components has become common practice, defining an acceptable state of the system is only possible to a very limited extent and thus exploratory analysis tasks by real human analysts complement the analysis process. However, this is a problem of scale since it is infeasible to manually inspect thousands of hosts without proper visual support for the tasks of gaining an overview, focusing and retrieving details on demand. In this paper we present a design study to enable visual support for monitoring large IP spaces. In particular, the presented system features 1) a scalable glyph representation in the style of a clock for giving an overview of the activity over time of thousands of hosts in the network, 2) subnet and port views for focusing the analysis to a particular subset of the data and 3) detailed pixel matrix visualizations for interpreting concrete traffic patterns. Furthermore, the tool's feedback loop, which is implemented through interaction capabilities, allows for retrieving new details, refocusing and enhancing of the overview.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.3.8 [**Computer Graphics**]: Application; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Network Security, Pattern Detection

1. INTRODUCTION

During the last few years an increasing number of viruses, trojans, worms and other malware have been circulating on the Internet infecting more and more computers. It will become an even greater challenge in the future to keep a network safe from all this anomalous traffic. Because of the possibility to download hacker scripts or to obtain information about certain security lacks from the internet, even non expert users are able to develop their own malware. This creates an unlimited amount of malicious software which makes it very difficult to keep every machine in a network secure.

After a computer has been hacked, the cyber criminal can manipulate the machine and cause unrepairable damages, including stealing personal data, sending spam or expanding his own botnet. Many times the ordinary user does not even realize that his computer has been hacked, which makes it even harder for the network administrator to monitor and secure the network. In the worst case, the malware can spread from this infected computer to other machines in the network causing widespread damage. Not being aware of the menace from the Internet, many computer users pay little attention to security updates or other defense mechanisms.

Detecting anomalous traffic in an entire company network is difficult because of two reasons. First, since the number of machines in a network grows at a rapid pace, many different hosts have to be monitored over time. Second, the amount of traffic leaving or entering the network grows relative to the number of new hosts. Thus, there is a need for network security tools helping the administrator to analyze the traffic. This massive amount of data cannot be effectively investigated by sequentially reading textual log files. Researchers and practitioners are aware of this fact and developed many different tools and concepts to apply filtering and visualization methods to this kind of data in the last few years. The goal is to support the administrator in dealing with this massive amount of data and in exploring anomalous traffic. Besides operationally monitoring real-time traffic to supervise a network, forensic analysis becomes an important aspect to reveal attack patterns and develop defense mechanisms against future attacks through diversifying malware aimed at circumventing traditional defense mechanisms. We thus believe that scalable visual support for forensic analysis tasks can complement currently used automated detection mechanism for a more holistic view on emerging threads in IP networks.

In this paper, we introduce the *ClockView* system with its different visualization techniques. The core contribution of this paper is the scalable visual pattern detection tool, which is capable of showing the temporal activity of thousands of hosts at once. This visualization builds upon the structural

VizSec '11, July 20, 2011, Pittsburg, PA, USA

properties of IP addresses belonging to subnets and a global prefix and therefore describes a two-level hierarchical data structure. Every visual item (host) shows temporal activity (traffic) as a small 24 hours clock.

Note that while showing the tool on real data, we only use anonymized *NetFlow* data in this paper to guarantee confidentiality towards our users.

The rest of this paper is structured as follows. Section 2 introduces a state-of-the-art analysis to outline important research results in this area and emphasizes the need for another network security tool. To understand the data gathering and preprocessing, Section 3 explains certain aspects of traffic flows and describes the underlying data structure of the ClockView tool. The software itself will be explained in detail in the Section 4. A short evaluation of the implementation is provided through a case study in Section 5. The last section summarizes and concludes the work and reveals future plans.

2. RELATED WORK

Information visualization techniques have successfully been applied to many different domains. One of the latest fields is network security with its massive amount of network traffic data. Since the last few years a lot of research in this area has been done and many tools were created to support the analysts in exploring the dataset. That is why this section only covers close related systems in either the examined dataset or the visualizations used.

A very similar approach for detecting patterns of anomalous traffic is realized by the tool NVisionIP [8]. The authors created a software to detect patterns with an overview visualization, showing an entire network of hosts in a 2D matrix divided into different subnets and host IP addresses. Every host is represented as a four pixel rectangle. The color of each rectangle codes the traffic of the host on different ports. Unfortunately, the visualization only shows one state of time at a glance. Anomalous behavior over time cannot be discovered on one sight. A more detailed perspective of the network is provided by the *Small Multiple View*, which uses two bar charts to visualize further information about the hosts. However, the overview is lost because only a limited amount of hosts can be displayed on the screen in this detailed way. To obtain more information, the analyst can dig deeper and investigate a single host by looking at its raw traffic data in the *Machine View*.

This approach was inspiring because of the way the network is monitored in a matrix visualization using small representations for every single host with the possibility to get details on demand. However, the way in which the hosts were displayed was not satisfying. With the aforementioned representation it was only possible to code a single parameter within each host (e.g. number of ports used). Therefore, glyphs would be a better way to display single machines in the network to have the possibility to code more parameters without loosing the overview. Krasser [7] and Pearlman [10] tried such an approach in their tools by using a glyph visualization in combination with network traffic data.

Krasser developed a 3D parallel coordinate plot in combination with glyphs. Two vertical axes show the source IP address and the destination IP address. For each transferred packet a line is drawn to connect the two axes. Color is used to distinguish between UDP and TCP traffic. Additionally, two glyphs are created for every occuring packet and each is placed on the height of the corresponding source and destination IP address near the axes. The glyph codes with its height the packet size. The distance of the glyph to the axes codes the time passed since the packet was transmitted. The analyst can navigate through the information space and zoom into interesting areas to get a closer look at the data. Additionally, the glyphs are clickable to reveal further information like port numbers or the protocol. Unfortunately the glyphs only code two parameters.

On the other hand Pearlman combined glyphs with a graph layout. Each glyph represents a node and codes the presence and amount of open services of the host in a pie chart. The size of the glyph visualizes the total amount of traffic while the size of the different regions of the glyph represents the relative amount of traffic for each service. Color is used to distinguish between the different services however there is a need to reuse some color because of the possible amount of services and the limited amount of color. Ring based temporal slices display changes over time, where the outer ring represents the most recent time slice. Two glyphs are connected if their services communicate with each other. The tool has scalability problems when the network gets too large or when too many different services are involved because of the limited space of the circle and the limited amount of different colors.

Tools like VIAssist [3], NFlowVis [2] or Rumint [1] try to provide different perspectives on the data with many kinds of visualizations. Graphs, parallel coordinates or scatterplots are used to display the data in multiple ways to reveal interesting facts or interrelations. With the help of certain interaction techniques like linking and brushing the analyst is able to investigate selected data points on different views to reveal their global behavior. Because of the explorative task of the network analyst it is necessary to provide him with multiple visualizations and interaction technqiues to give him the possibility to investigate the data in different ways.

In summary, the state-of-the-art analysis has shown that there is a need to improve the host representation in a way that more parameters can be displayed on one sight without loosing the overview when monitoring an entire network. We are not aware of any other visual pattern detection tool showing the activity of thousands of hosts over time in an as scalable way as ClockView. Following an overview, focus and details on demand approach, it is thus necessary to provide the analyst with the possibility to investigate the data in a more detailed way to foster deeper understanding.

3. DATA PERSPECTIVE

The tool described in this paper uses *NetFlows*[1] as data source. *NetFlows* are located one layer above the packet captures and are collected primarily on routers and switches [9]. A single flow can contain information about multiple packets passing through the router. Information about packets with identical source and destination IP address, the same protocol and ports within a certain timeframe are summarized into one flow.

A server with different software solutions like for example *flow-tools*[2] was installed to collect the data. Since the UDP protocol is responsible for the export in real-time, the flows

[1] http://www.cisco.com/go/netflow

[2] http://www.splintered.net/sw/flow-tools/

are stored to RAM every five minutes to avoid a possible data loss at this early stage. In the network used for our research there are about 300 million *NetFlows* on a normal business day.

This massive amount of data has to be transformed in a file format adequate for a fast import in the PostgreSQL database. Therefore, comma separated text files are a good choice. Due to hardware limitation issues it was only possible to import the data once every day. The import rate could be increased by updating the hardware and investing more time to speed up the preprocessing. For each daily import a new table in the database is created to improve the build speed of the indices.

To use the data for scientific purposes and to maintain the privacy of the network users, it was necessary to anonymize the data before importing into the database. This step can be discarded for operational usage thus improving the performance. To provide the interactivity of the developed visualization tool even with very complex queries, several aggregated views were created which are stored in a separate table. These views prepare an interface where the data which is necessary for certain queries can be accessed very fast thus improving the performance of predefined queries dramatically.

4. CLOCKVIEW

4.1 Motivation

To detect abnormal traffic over time automated algorithms and visualizations must be combined to profit most of the computational power of the computer and the visual perception of the human. A scalable visualization helps the analyst to understand the outcome of the algorithms and to interpret the results efficiently. In order to detect conspicuous traffic changes over time and to investigate the findings in more detail, the tool ClockView was developed. ClockView is able to adequately display suspicious traffic patterns on an hourly basis or to compare the traffic volume over many days. In the current version, ClockView satisfies different use cases:

1. **Detecting Suspicious Traffic:** The analyst monitors a whole network and tries to detect suspicious traffic over time concerning the whole network or single hosts. Different filtering options help to find an interesting pattern. After selecting a host the user receives additional information like for example connections to other hosts or the geo-location.

2. **Forensic Analysis:** The analyst combines historical data with the most recent one in order to detect irregularities. The deviation in traffic over time will be displayed in the same way as the overview.

3. **Data Fusion:** The analyst extends the data with other sources. These additional sources will support the investigation with further insightful information.

4. **Feedback Loop:** The analyst can save and use his knowledge gained from previous investigations for future analysis.

4.2 Workflow

Due to the large amount of data available, Shneiderman's information seeking mantra "Overview first, zoom and filter, then details-on-demand" [11] was kept in mind while developing the workflow (Figure 1) of ClockView. ClockView's visualizations, to exploratively analyze and monitor the network traffic, can be grouped into three categories: At first the *Network Overview* and the *Subnet View* (Figure 3) with different glyphs and layout options provide the network analyst with an overview of the internal network. Internal hosts can be selected to further zoom into the data. Additionally, interesting external IP addresses can be chosen from pregenerated lists of blacklisted or scanning hosts. In the second part (the *Focus*), the *Host Matrix* and the *Parallel Coordinates View* (Figure 5) can be used to investigate the traffic of the selected host. In these visualizations a second host can be selected to enable the third category. This third category consists of the *Port Matrix* (Figure 6), which provides the user with information about the whole traffic between these two hosts as details-on-demand.

Two additional *feedback loops* allow the user to carry out an iterative analysis. The first *feedback loop* is realized by a global filtering system, accomplished by the concept of *Dynamic Querying*. Different ports, protocols and traffic types (incoming, outgoing or both) can be chosen. Filters selected in one representation are also applied to all other visualizations and therefore give the user the ability to easily refocus on the other views. The second *feedback loop* is realized through a pattern management interface. The term pattern refers to a general condition that a host or the connection between two hosts match, for example all machines with traffic on port 22 (SSH). The user is visually supported in defining such patterns to find similar hosts. For the ease of use, a built-in database query template can be used to generate a pattern based on the currently selected filters and hosts. Since the most common patterns can be already expressed by the global filtering system, this option is better suited for advanced users, who want to modify these database queries to express more complex patterns or build arbitrary queries to the database itself. With the pattern management, it is also possible to integrate external data sources, like blacklists (e.g. DShield[3]), data collected from honeypots, or alarms of an intrusion detection system (e.g. Snort[4]).

To keep track of the current selection of hosts and filters, they are shown in the upper left. Additionally, further information about the IP address can be found there. These include the current hostname, the country according to a geolocation database and some statistics, like the total number of *NetFlows* and connections to distinct other hosts.

4.3 Glyph Visualization

To get a global picture of the servers and workstations used in the network, it is useful to visually encode each host individually in the *Network Overview*. The hosts are represented in a way the user can easily notice, if a specific machine's behavior matches more a server with 24 hours of traffic or a client with only traffic on the working hours. Therefore, we want to show all internal hosts with their traffic at a granularity of one hour for a timespan of one day.

[3]DShield Blacklist, http://www.dshield.org
[4]Snort, http://www.snort.org

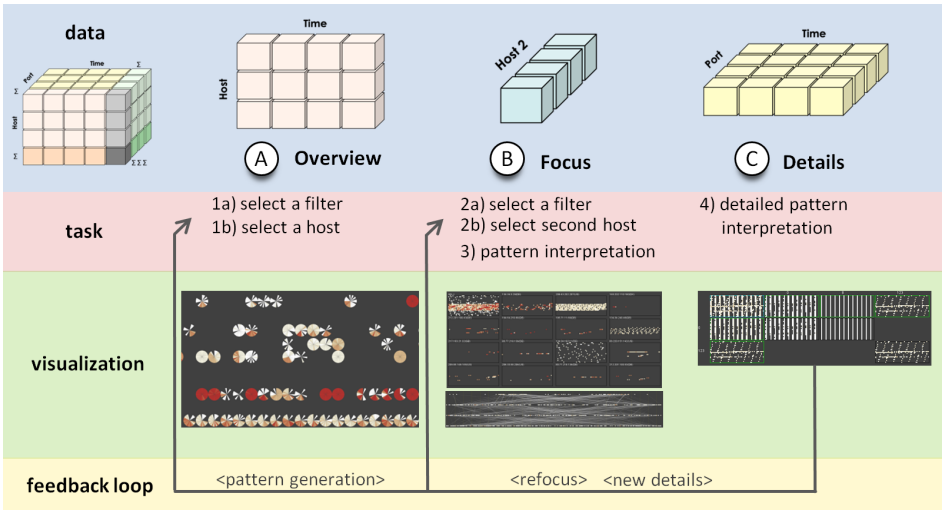

Figure 1: Analysis workflow with ClockView

For this purpose we need to display up to 65536 (256*256 possible IP addresses for a /16 network) time series, each with 24 (one per hour) data values. This leads to a maximum of 1572864 data points. We implemented 4 different versions (Figure 2) to visualize one time series.

The first is a simple line chart arranging the hours on the x-axis and the amount of traffic on the y-axis , which is a well-known visualization technique and therefore easy to understand. Changes within one time series can be well detected. However, given this large amount of time series, it is not possible to assign at least 1 pixel per data value in width for the line chart on a normal screen resolution. The comparison between different time series is difficult, because the line charts are mostly far apart from each other and not all of them can be aligned side by side.

The second representation is a bar chart, where the traffic is double encoded to the height and with a colorscale ranging from white (low traffic) to red (high traffic) of the bars. Regarding the space in width the same problem occurs as with the line chart. However, they were better comparable because of the usage of color. The main problem of the line and bar chart is, that they rely on position - the farther apart they are, the harder they are to be compared.

Based on the above, we used a more space-filling pixel-oriented visualization [6] and encode the amount of traffic only with color. In this third representation every hour is represented by a pixel/rectangle. Hours without traffic remain in the background color, to perceive the clear cut between hours with traffic and no traffic. The rectangles of one time series are arranged line-wise in a 4 times 6 matrix. Even though they are separated by spacing, hosts with irregular activity can hardly be distinguished. The comparison between time series is better than the comparison with the line and bar chart, since for every data value there is more space available and the amount of traffic is no longer represented by the position.

The fourth representation is a glyph in style of a clock. Each circular glyph is subdivided into 24 segments, each of them showing the traffic of one hour encoded with color. 0:00 o'clock is at the top, 6:00 o'clock at the right side, 12:00 o'clock at the bottom and 18:00 o'clock at the left

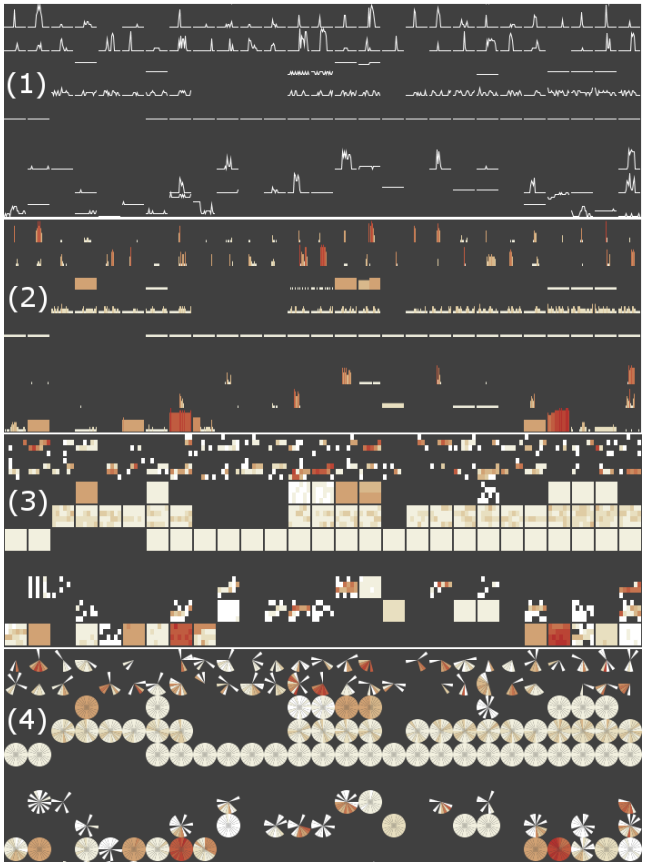

Figure 2: The same time series in four different representations (1) line chart (2) colored bar chart (3) pixel matrix (4) glyph in style of a clock

side. As a clock metaphor is used here, this segmentation is more intuitive as the segmentation into rectangles, even if the clock is transformed from 12 to 24 hours. Also the natural order of time is better preserved, since there are no line breaks between the data points. The time representing segments are not only at the same position for every host, but also have the same orientation. Corresponding hours of different hosts are displayed in parallel and thus at a glance can be recognized as group. Since the separation between the glyphs is already achieved due to the circular shape, no additional spacing has to be added. Because of this, the glyph is more space-efficient on smaller screen resolutions.

The amount of traffic is represented by a fixed diverging color scale from blue (negative, only used for comparison showing a decrease in traffic) over white (0) to red (positive). Due to the fixed color scale hosts remain comparable on different days. Otherwise a host with the same amount of traffic on different days could be perceived totally different. However, a drawback of this design choice is that the exact value the color represents is not visible.

Due to the above mentioned reasons, we think, that the glyph is the most appropriate way for displaying the large amount of time series, although the other representations have their advantages, too.

4.4 Network Overview

Different layout options are available to arrange the glyphs in the *Network Overview*. In the first layout the glyphs are represented in a matrix. The subnets (in this case the 3rd byte of the IP address) are arranged on the y-axis and the individual hosts (4th byte of the IP address) are arranged on the x-axis. This positioning was chosen because subnets without traffic can optionally be removed from the matrix to better fit the available screen resolution, since computer screens are mostly oriented horizontally. With this layout the user can not only directly locate a certain IP address of interest, but also see trends occurring within a subnet (horizontal) or on the same 4th byte of the IP address (vertical). In the second layout the glyphs are arranged recursively. The first dimension is the 3rd byte of the IP address and the second dimension is the 4th byte of the IP address. Since this layout minimizes the distance between IP addresses in the same subnet, trends within a subnet can be better spotted in this layout. The third layout arranges the glyphs in order of their total traffic within the network or subnet-wise. The top talkers of the whole network or within a subnet can be easily spotted here. This layout is more space-efficient, because gaps from hosts without traffic are discarded. Unfortunately there is no direct mapping of an IP address to the position on the screen.

By default, the traffic of the current day is mapped to the glyph, but the user has two additional options to change the information that is represented by the glyph. The first option, which is the coefficient of variation of the previous days (3, 5 or 7 days), gives a clue about how stable the traffic is. To find anomalous traffic, we combined these 2 measures. Therefore, the second option for every hour is computed as follows: $\frac{x-mean}{stddev+1}$, where x is the value of the current day, *mean* is the average and *stddev* the standard deviation of the previous days. To avoid a division by zero an alpha value of 1 is added to the standard deviation. The result is a value, which indicates the relative change in comparison to the normal behaviour on the previous days. Additional options can be used for filtering. They display the absolute traffic of the currently selected day only for those data values (hours), which had either no traffic or exceed the maximum traffic of the previous days.

For an easier navigation, a tooltip with the IP address of the host appears on a mouse-over over the corresponding glyph. Zooming possibilities allow the user not only to adapt the size of the visualization to his screen resolution or personal preferences, but also make it easier to choose a clickable glyph. By selecting a glyph the visualization is updated and all connections within the internal network to the chosen machine are shown by lines connecting the glyphs. Optionally, each communication line between two machines in the network can be displayed on top of the visualization. These lines form an internal graph of the network. Connections with traffic below a user defined threshold can be discarded. The user is able to define the transparency of the lines. If the internal graph is visible, the connections of the selected host will be highlighted additionally (Figure 4).

Several filtering methods can be applied to the visualization like for example the global filters for port, protocol and traffic. They are located on the top right of the screen and represented in different ways. A table contains the amount of traffic and a checkbox for each used source- and destinationport. By selecting a checkbox the different views will be constrained to show only the traffic on these specific ports. Furthermore, the user can decide to see only the activity of the hosts on a single protocol by choosing the corresponding entry in a dropdown list (e.g. ICMP, TCP, UDP). To reduce the amount of visible hosts the analyst is able to focus only on those machines having incoming as well as outgoing traffic. IP addresses which receive traffic but are not assigned will be discarded. The filters set are not only affecting the glyphs, but also the internal graph on top of the view.

In addition, hosts can be filtered out by choosing one or more of the predefined patterns in the list on the lower right. Each of them can be selected positively or negatively to show the machines, which either match or do not match with the given characteristics. The list of manually named patterns displays the percentage of the currently visible hosts that match the specified properties. The navigation in the list is additionally improved by the use of a Visual Scent [13] in form of a bar chart (Frame 7 in Figure 3).

4.5 Focus

The second group of visualizations is available once the user has selected a glyph from the *Network Overview* and focuses on the traffic of this specific device. The *Subnet View* (Frame 2 in Figure 3) on the left side of the screen provides the user with a rough overview of the connections to that host. While also visible from the *Network Overview* and the *Port Matrix*, the *Subnet View* links these visualizations to each other. The *Host Matrix* view replaces the overview visualization in the middle of the screen and displays the activity between the previously selected host and its counterparts in more detail. The filtering options can be adopted to this representation as well. Besides the global options, a custom one is available to filter according to an IP address range. This functionality was accomplished within the *Parallel Coordinates View* (Frame 2 in Figure 5). Applied filters do not only affect the *Host Matrix*, but also the *Subnet View*

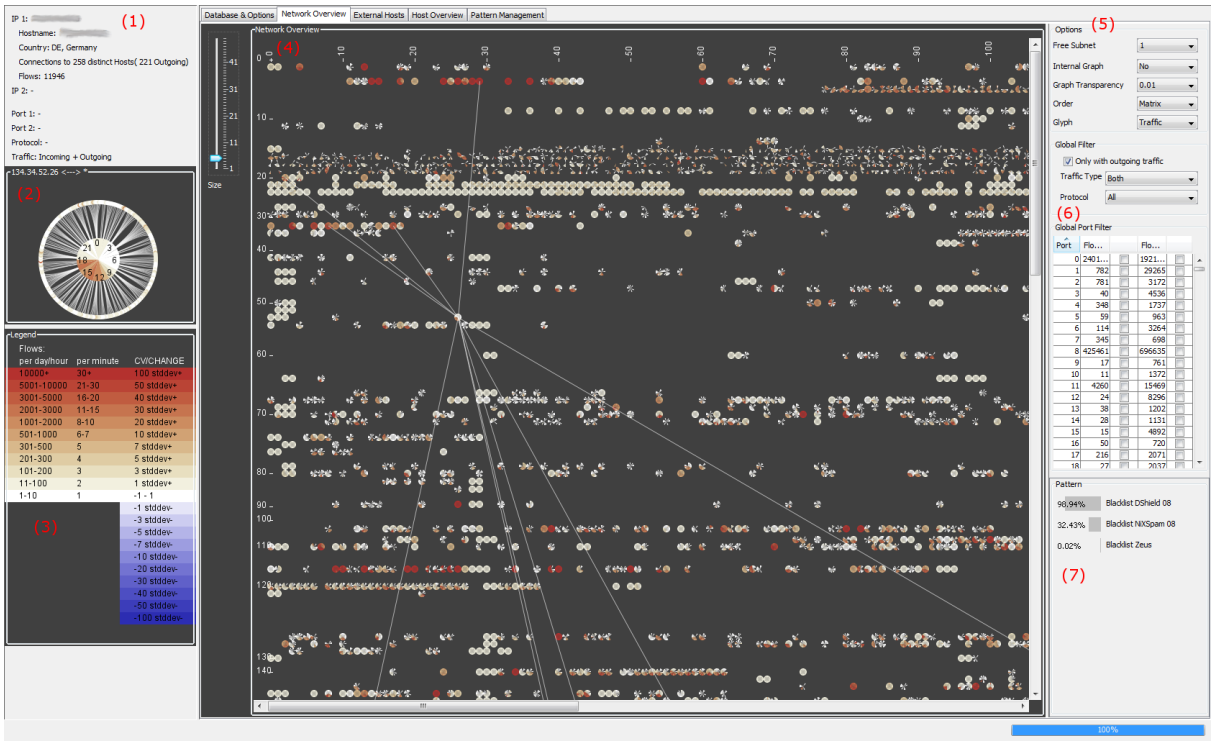

Figure 3: Graphical user interface of ClockView: (1) Host Information, (2) *Subnet View*, (3) Color Legend, (4) *Network Overview*, (5) Options, (6) Global Filters and (7) Patterns

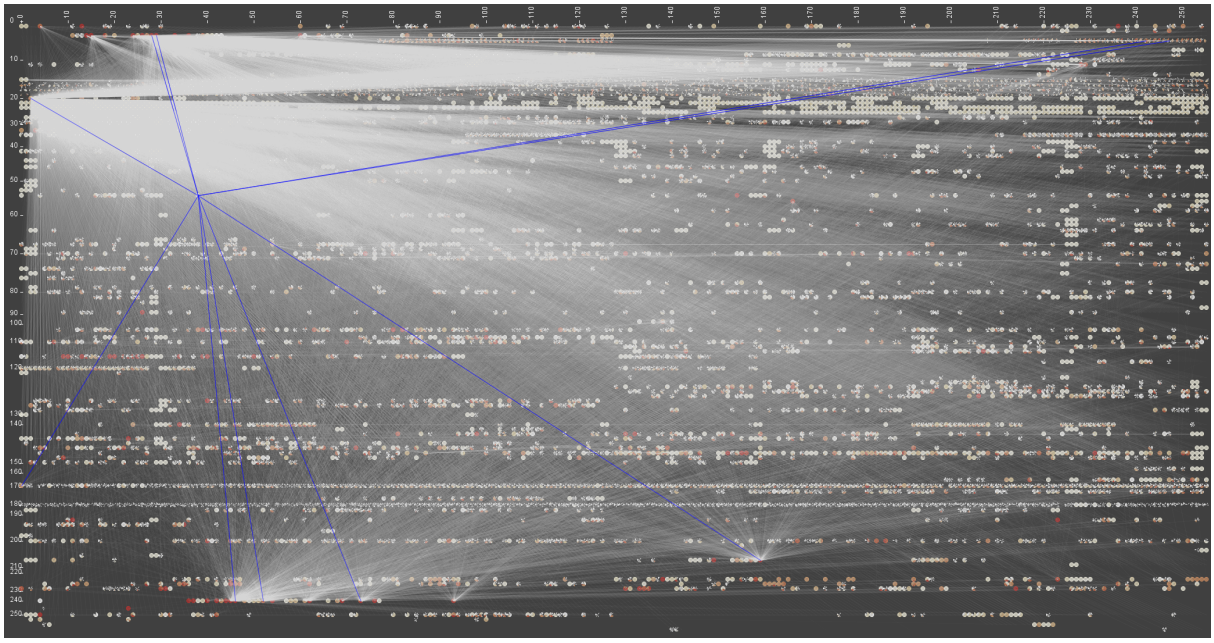

Figure 4: Internal graph on top of the *Network Overview*. Communication lines of the selected glyph are additionally highlighted.

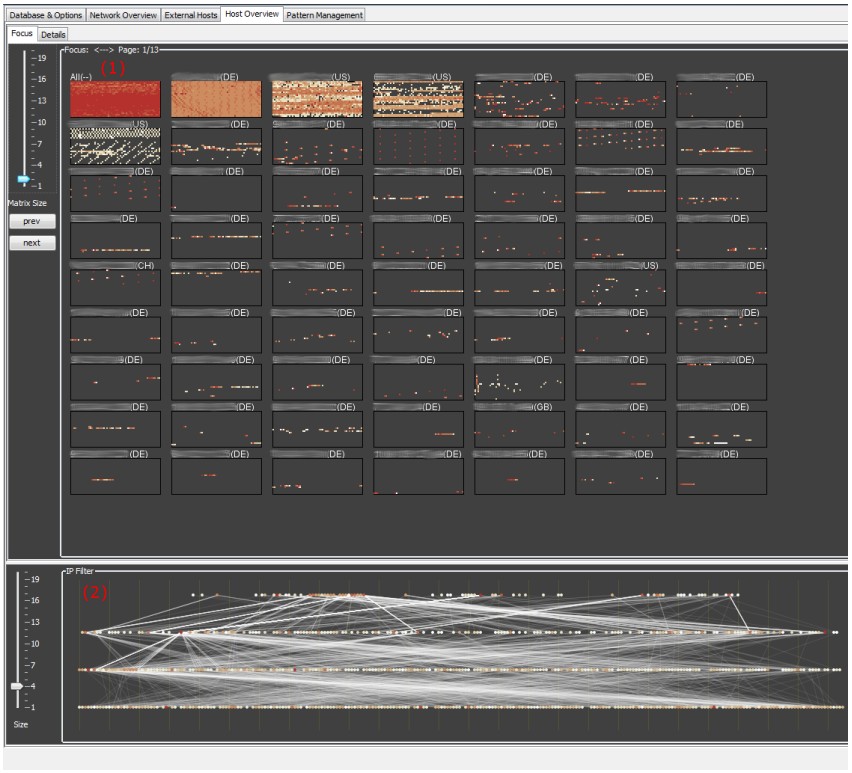

Figure 5: Two *Focus* visualizations replacing the *Network Overview* in the middle of the screen. (1) *Host Matrix* (2) Parallel Coordinates View

and the selectable port and IP address range filters. For example, if the user decides to constrain the views to a certain IP address range, the statistics on the port table will be updated. Only the ports and the amount of traffic within the chosen IP address range are shown. Selections made in one view are reflected in the other views using *Linking & Brushing*.

4.5.1 Subnet View

The *Subnet View* displays every host communicating with the previously selected glyph (Frame 2 in Figure 3). This glyph is located in the center of the visualization and is connected with its counterparts via lines. These connection lines are bundled using the Edge Bundling Technique [4] according to the first Byte of the hosts' IP addresses. The approach was inspired by the network security tool FloVis [12] to improve the layout of similar IP addresses. The counterparts themselves are equally distributed on an additional outer ring around the glyph and are represented by a small colored circle. The color codes the amount of traffic between the two communicating hosts using the same fixed color scale as in the *Network Overview*. To gain further information, a mouse-over over any host provides the analyst with the exact IP address, the hostname and the country according the geolocation database. A further host can be selected for the *Port Matrix* view by clicking on the corresponding circle. This second host, even if it was chosen in another visualization, will be highlighted additionally. The Subnet View has scalability problems when there are too many connections resulting in many items placed on the outer ring. This problem can be solved by applying some filters thus reducing the number of counterparts. Nevertheless the user is provided a rough overview of the distribution of the communicating hosts.

4.5.2 Host Matrix

The *Host Matrix* is the most detailed visualization of the *Focus* category. While the glyphs in the *Network Overview* and the *Subnet View* represent the traffic at a granularity of 1 hour, the *Host Matrix* uses a granularity of 1 minute. Visualizations at a finer grade do only make partially sense, since a *NetFlow* consists of one or more packets itself and therefore the timestamp is not totally accurate. In the *Host Matrix* the traffic of the chosen device is subdivided by the IP address into multiple time series. An additional one shows the whole traffic of the chosen device. The time series are arranged as *Small Multiples* in a matrix.

With only one dimension in this matrix, the time series can be ordered by their IP address, their country or the total amount of traffic. Because of the possible high number of hosts pagination was used to provide the overview. However, in a typical workflow the user would first apply some filters, to reduce the number of available and interesting hosts and therefore pages. The filters also affect the traffic shown by the times series, since in the *Network Overview* patterns are available to filter out certain hosts. Every time series is represented by a pixel matrix visualization [6], where each pixel displays one minute of the day. For this purpose 60 minutes are shown in one row, resulting in a total of 24 rows for one day. The color of the pixels codes the amount of traffic. Since the granularity is lower in this view, the color scale represents different values as in the *Network Overview*.

Especially regular time patterns can be easily recognized. Zooming possibilities allow the user to reveal more details. It is not only possible to choose an IP address for the *Port Matrix* in this view, but also directly switch the visualizations of the *Focus* group to another chosen device. Like in the *Subnet View*, a selected host will be highlighted in the *Host Matrix*.

4.5.3 Parallel Coordinates View

The *Parallel Coordinates View* uses Parallel Coordinates [5] to display all connected hosts. Each of the 4 bytes of an IP address is shown on one axis. On each axis are 256 data points. A connection line between each of the four axes symbolizes a single host. The amount of traffic is represented on every data point by the same fixed color scale as used in the *Network Overview* and appears as tooltip on every data point as well. With this visualization the user should get an insight to the structure of communicating machines. The second utilization of this view is the possibility to filter by an IP address range. Since every data point represents one structural part of an IP address, this can be achieved by clicking on one or more of the data points.

4.6 Details: Port Matrix

The *Port Matrix* (Figure 6) shows the detailed activity between two machines. As in the *Host Matrix*, a single time series is represented by a pixel matrix with the same color scale. The traffic is subdivided according to the ports used. A variation of *Dimensional Stacking* was applied to align the *Small Multiples* of time series. The outer dimensions are defined by the port combination, the inner dimensions represent the time in the pixel matrix. In the outer matrix, the ports of the first host are arranged on the y-axis, the ports of the second host on the x-axis. Additionally, the sum of all time series of each row is shown on the left side of the row and the sum of all time series of each column is shown on top of the column. Each sum represents the whole traffic on the corresponding port to reveal possible time patterns that would otherwise be distributed over many single pixel matrices. The aggregated traffic between the two machines can be seen on the upper left corner of the matrix. On a mouse-over a tooltip appears with additional information like the official IANA port assignments[5] (e.g. SSH for port 22) or information about known malware using a certain port. By clicking on one of the pixel matrices, the user directly selects the corresponding port filter and can therefore easily refocus to this port (-combination) on the preceding views.

5. CASE STUDY

To evaluate the software for operational usage the case study deals with different use cases to monitor a whole class B company network and detect anomalous traffic. Therefore, the tool was used in different ways to exemplify the variety of the possibilities provided.

5.1 Monitoring an entire company network

This case study describes a typical workflow for monitoring one day of network activity. After selecting the desired day in the database, we switch to the *Network Overview* visualization to start the analysis. In this view, as described

[5]http://www.iana.org/assignments/port-numbers

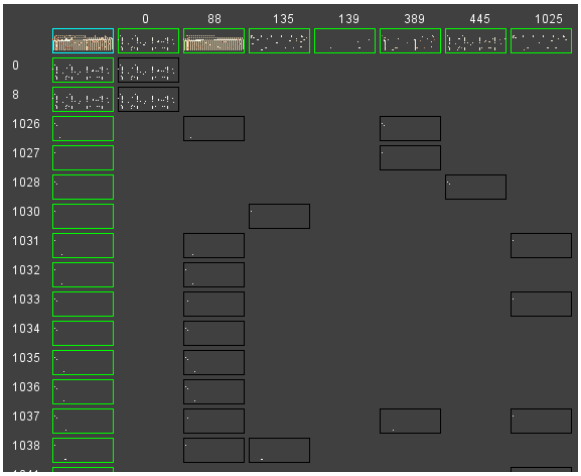

Figure 6: *Port Matrix*

in section 4.4 in more detail, the daily activity of the entire company network is visible at a single glance (Figure 3). Top talkers as well as different patterns are easy to spot and with some background knowledge of the network structure easy to interpret like for example the continuously cluttered red clocks within a certain subnet range (Figure 7). This fissured pattern is caused by assigning a dynamic IP address to each computer establishing a connection within the wireless lan network to the internet. As a result, a single glyph, although representing a unique IP address, can possibly display the traffic of many different computers. Most of these computers belong to students using the wireless lan connection on an irregular basis.

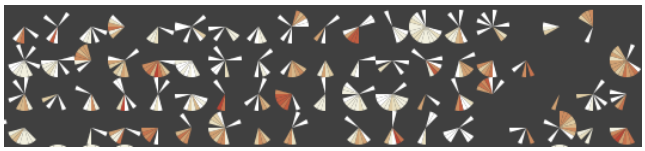

Figure 7: Typical glyph pattern of hosts with dynamic IP addresses using the wireless lan connection in the university

With some background knowledge most of the patterns can be easily explained. To spot abnormal behavior without additional knowledge about the network the tool provides helpful interactive features. As an example, we select the item “Change 5 days” in the dropbox next to the label named “glyph”. This option compares the traffic of the current day with those of the five previous ones and displays the result in the glyph. As expected, most of the glyphs are colored white thus signalizing nearly no change, except for a partial red pattern in one single subnet (Figure 8). To take a closer look at the single glyphs we enlarge the visual representations by zooming in this exact area. With the additional space for each circle the traffic distribution over time is getting more obvious. Basically on the second half of the day the amount of traffic rises. It seems that some new machines have been added to the network causing extra traffic. This is suspicious because the monitored dataset was a Sunday where there is no regular daily work in the university. After investigation, we discovered that the cor-

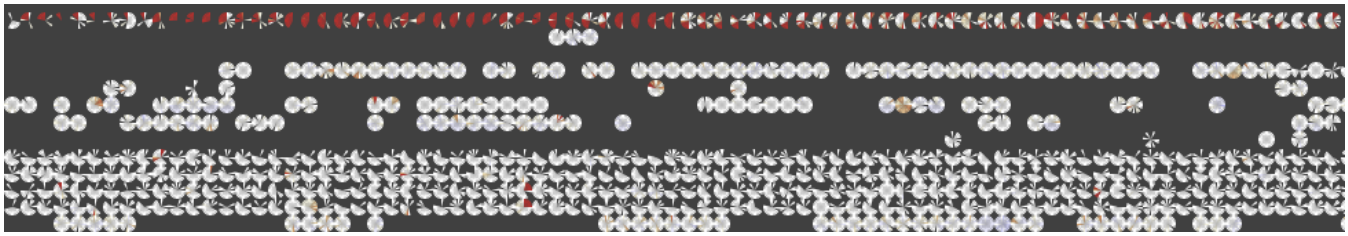

Figure 8: Subpart of the *Network Overview* showing a relative increase of traffic compared to the previous days

responding subnet of the university is assigned to the vpn connections. A computer connecting to the university from an external network gets an IP address in this specific subnet. With this additional information the suspicious pattern can be explained as a common occurrence.

5.2 Integration of external data sources

Since additional knowledge is often crucial to detect anomalous behaviour, the traffic of the university's network is matched with different blacklists. For this purpose we define different patterns. The condition of the pattern is expressed in a way, that internal hosts are required to have traffic with at least one of the IP addresses on the corresponding blacklist.

The first blacklist we choose is a very general one with all kinds of threats from DShield. As an interesting fact, nearly every computer (about 98 per cent) has some kind of activity with at least one blacklisted IP address. Fortunately, the only ones without activity are the hosts in the so called demilitarized zone. Since this list is very general and often the matching results only from a scan, the findings can be improved by defining a certain threshold of minimum traffic in the pattern.

The second blacklist we choose is a more specific blacklist of known *ZeuS Command & Control*[6] servers. Once a computer is infected with the *ZeuS* trojan, it becomes part of a botnet and communicates with its *Command & Control* server on a regular basis. The matching with the blacklist reveales, that there is one computer within the university's network with activity to one of the blacklisted IP addresses on several consecutive days. The inspection in the focus & detail visualizations as detailed in Figure 9 shows that the traffic to the specific server is indeed on a regular basis. In the pixel matrix actually two regular communication patterns can be distinguished between the infiltrated host and his master server. The first shows activity about every 5 minutes and the second one about every 20 minutes. Because of the additional knowledge gathered by using the blacklist, we think that this host has been hacked and should be manually checked. This example shows the usefulness of additional information and the integration of external data sources. Without this knowledge this specific host could not have been found with only the data generated from the *NetFlows*, since the host has shown no additional extraordinary traffic besides the periodic communication to his master. Since periodic communication is by no means a sign of an intrusion by default (e.g. a mail program checking for new arrived mails every 5 minutes), it also cannot be used in general to detect anomalous traffic.

[6]abuse.ch ZeuS Tracker, https://zeustracker.abuse.ch

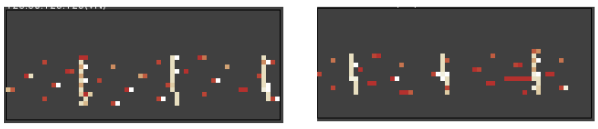

Figure 9: Pixel matrices of a hacked host communicating with its *Command & Control* server on two consecutive days in a 60min x 24h pixel matrix

6. CONCLUSIONS

This paper presented the network security tool ClockView. ClockView displays the daily activity of a whole company network in a scalable glyph based visualization. To provide this scalability, every single IP address is broken down to its subnet and host identifier to perfectly fit in a matrix layout. Each device is then represented by a round glyph subdivided into 24 different areas. This 24 hour clock metaphor shows the activity of a whole day by color coding each slice according to the amount of traffic at the corresponding hour. After detecting something suspicious in the overview the analyst is able to dig deeper by investigating certain areas in a more detailed view. Different ordering, layout and analysis algorithms, like the traffic change of a single host over many days, support the user in exploring the dataset. If he reveals something interesting he is able to safe this discovery with its characteristics in a pattern to enable a *feedback loop*. This pattern can then be applied to all the other visualizations to filter the views for the previously detected findings. Additionally, each predefined pattern can be used with other datasets to quickly scan the network at different times.

The tool was tested with real anonymized *NetFlows* of a whole class B IP network to assure its operational suitability. The use cases described in chapter 5 verify the applicability of the software in a real environment with actual traffic data. This distinguishes ClockView from other tools developed only for research purposes.

To further push the operational usage, our future plans are to design a user study to improve the usability of the software and therefore simplify the explorative and analytical tasks of a network administrator. Furthermore, we intend to support more historical views to gain different perspectives on the data, since these are often crucial to detect anomalous traffic. Another issue we want to address is the generation of patterns, which support better constrains in terms of time. We also plan to implement an adequate similarity measure between time series. Once this is completed

hosts in the *Network Overview* could for example be laid out by a clustering algorithm based on this similarity measure.

7. ACKNOWLEDGMENTS

We thank the IT Service Centre at the University of Konstanz for enabling this research project by making anonymized NetFlows available to us. The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 257495.

8. REFERENCES

[1] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. Copeland, M. Ahamad, H. Owen, and C. Lee. Countering security information overload through alert and packet visualization. *IEEE Computer Graphics and Applications*, pages 60–70, 2006.

[2] F. Fischer, F. Mansmann, D. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. In *Visualization for Computer Security: 5th International Workshop, Vizsec 2008, Cambridge, Ma, USA, September 15, 2008, Proceedings*, page 111, 2008.

[3] J. Goodall and M. Sowul. VIAssist: Visual analytics for cyber defense. In *Technologies for Homeland Security, 2009. HST'09. IEEE Conference on*, pages 143–150. IEEE, 2009.

[4] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12:741–748, September 2006.

[5] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[6] D. A. Keim. Designing Pixel-oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1):59–78, January–March 2000.

[7] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *Proceedings of the 6th IEEE Information Assurance Workshop*, volume 142. Citeseer, 2005.

[8] K. Lakkaraju, W. Yurcik, R. Bearavolu, and A. Lee. NVisionIP: an interactive network flow visualization tool for security. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2675–2680. IEEE, 2005.

[9] R. Marty. *Applied security visualization.* Addison-Wesley, 2008.

[10] J. Pearlman and P. Rheingans. Visualizing network security events using compound glyphs from a service-oriented perspective. *VizSEC 2007*, pages 131–146, 2008.

[11] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society, 1996.

[12] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh. Flovis: Flow visualization system. *Conference For Homeland Security, Cybersecurity Applications & Technology*, 0:186–198, 2009.

[13] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13:1129–1136, 2007.