Large-scale Network Monitoring for Visual Analysis of Attacks

Fabian Fischer, Florian Mansmann, Daniel A. Keim, Stephan Pietzko, and Marcel Waldvogel

University of Konstanz, Computer and Information Science,
78457 Konstanz, Germany
{Fabian.Fischer,Florian.Mansmann,Daniel.Keim,
Stephan.Pietzko,Marcel.Waldvogel}@uni-konstanz.de
http://infovis.uni-konstanz.de

Abstract. The importance of the Internet and our dependency on computer networks are steadily growing, which results in high costs and substantial consequences in case of successful intrusions, stolen data, and interrupted services. At the same time, a trend towards massive attacks against the network infrastructure is noticeable. Therefore, monitoring large networks has become an important field in practice and research. Through monitoring systems, attacks can be detected and analyzed to gain knowledge of how to better protect the network in the future. In the scope of this paper, we present a system to analyze NetFlow data using a relational database system. NetFlow records are linked with alerts from an intrusion detection system to enable efficient exploration of suspicious activity within the monitored network. Within the system, the monitored network is mapped to a TreeMap visualization, the attackers are arranged at the borders and linked using splines parameterized with prefix information. In a series of case studies, we demonstrate how the tool can be used to judge the relevance of alerts, to reveal massive distributed attacks, and to analyze service usage within a network.

Key words: visual network monitoring, visualization for network security, large-scale netflow analysis

1 Introduction

The increase of network attacks in terms of coverage, intensity, and aggressiveness is one of the most difficult challenges for network administrators today. A major part of these developments is to be attributed to so-called *Botnets*, which play a critical role in large-scale attacks [1]. In particular, more and more attacks focus on corporate and governmental networks with the goal of stealing confidential information or blackmailing companies whose business model depends on uninterrupted availability of their business and services. These facts point out the need for software to effectively and efficiently analyze network traffic both in real-time and for forensic purposes. The latter analysis can be especially useful for discovering compromised hosts within the local network.

Since the analysis of flow data in current analysis systems is only supported to some extent, we propose a novel analysis system called *NFlowVis* with the goal of enabling quick visual insights into communication patterns. The system is capable of storing NetFlow data of large systems, linking these flows to alerts from intrusion detection systems or public warnings, and to visualize flows between external and internal hosts. Using a TreeMap visualization, we depict the local network infrastructure emphasizing high traffic subnets. On top of this visualization, Splines in selected colors are utilized to connect the external host with the local communication partners, thereby revealing insight into communication patterns of malicious and legitimate network traffic.

To protect the privacy of our network users, we anonymized all IP addresses used while maintaining the grouping according to the higher level prefixes. Therefore, conclusions about the usage of a particular hosts, either internal or external ones, cannot be drawn from the displayed figures.

The remainder of this paper is structured as follows: Section 2 discusses related work, Section 3 introduces our *NFlowVis* application, Section 4 presents real-world case studies, Section 5 discusses the tool's applicability and scalability, and Section 6 summarizes our contributions.

2 Related Work

Visualization for computer security is a relatively young research field. While substantial research has been conducted in the field in the last few years, for brevity this section will focus on visual network traffic monitoring and discuss the roots of the used visualization concepts.

In the Open Source community, there are two popular tools: *NfSen* [2] and *Stager* [3]. Both tools comprise web frontends to display aggregated information about previously captured netflows. In the backend, database management systems enable efficient access to detailed information and efficient generation of aggregated reports. For visual analysis, both systems use line charts for displaying temporal overviews of network system load. While Stager only stores highly aggregated data, NfSen reverts back to the original flow data for detailed analysis.

Since network monitoring is particularly important for the health of the commercial network infrastructure, there exist a multitude of commercial systems. In contrast to the previously discussed tool, commercial systems such as *IBM Aurora*[1], *NetQoS Reporter Analyzer*[2], *Caligare Flow Inspector*[3], and *Arbor Peakflow*[4] often include methods for intrusion detection in which generated alerts can be examined through interactive reports. However, the used statistical charts and diagrams only scale to a limited number of alerts or highly aggregated information.

[1] http://www.zurich.ibm.com/aurora
[2] http://netqos.com/solutions/reporteranalyzer
[3] http://www.caligare.com/netflow
[4] http://www.arbornetworks.com

Visualization approaches in network monitoring aim at supporting the system administrator in the exploration of network traffic by means of interactive visual displays. *NVisionIP* [4], for example, enables visual pattern recognition and drill-down functionalities to inspect suspicious machines. *TNV* [5] is a network traffic visualization tool focusing on temporal aspects by means of a time versus internal host matrix, which details traffic flows for each host and links the external communication partners on the side. The home-centric network view of *VISUAL* [6] is probably closest to our proposed visualization since a matrix showing all internal hosts in the center is linked to external communication partners using straight connecting lines.

In contrast to this work, we made two major conceptual changes:

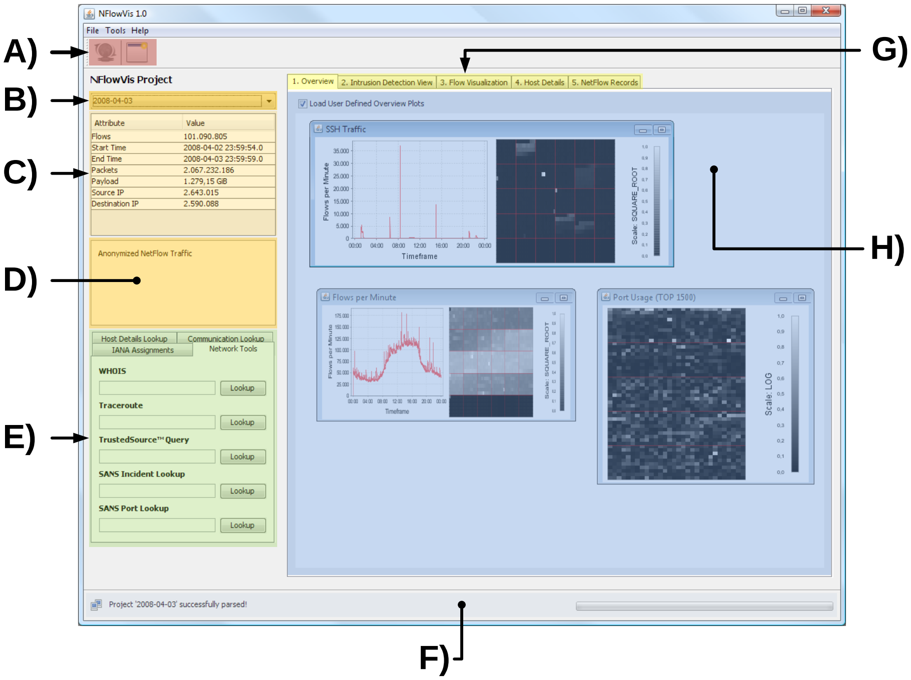
a) Instead of using a matrix view for the internal hosts, we employ a *TreeMap* [7] visualization, which hierarchically maps the monitored network infrastructure to prefixes of various granularity. Unlike in our previous work [8], high-load entities are thereby enlarged.
b) Rather than using straight lines to link the communication partners, we employ *Hierarchical Edge Bundles* [9] to visually group related flows, and thereby avoid visual clutter.

While we visualized flows using Hierarchical Edge Bundles with both start and end point within a TreeMap visualization in an earlier work [10], the work presented in this paper explicitly focuses on a home-centric network view, which represents the local IP prefixes or addresses in a TreeMap and places the external hosts at its border.

3 Visual analysis of attacks

Keeping the general workflow of a network analyst in mind, we developed *NFlowVis* to interpret the relevance of network security alerts. The system supports this full workflow through its five analysis views with a general network *overview*, an integrated *intrusion detection view*, the *flow visualization* of attackers' connections, a detailed *host view*, and the full *NetFlow records* of the specified communications as the most detailed view. In the graphical user interface these views are represented through several tabs to emphasize the drill-down and filtering process. Fig. 1 describes the design of NFlowVis: the connection settings to the database server and project creation wizard (A), project selection (B), key data of the selected project (C), textual description (D), fast access to external tools and internal queries to retrieve host details (E), current status and progress of operations (F), and the previously mentioned data exploration views ordered according to the levels of details (G).

Within the *overview* tab, the system provides several user-defined plots (H). With the help of these graphs the analyst is able get a rough overview of the actual network situation and utilization detailing the aggregated traffic and port usage within the whole network. To visualize these time series we use line charts



Attribute	Value
Flows	101.090.805
Start Time	2008-04-02 23:59:54.0
End Time	2008-04-03 23:59:59.0
Packets	2.067.232.186
Payload	1.279,15 GiB
Source IP	2.643.015
Destination IP	2.590.088

Project '2008-04-03' successfully parsed!

Fig. 1. User interface of the NFlowVis system showing the annotated main view (left).

and grouped line-wise pixel arrangements. The use of both visualizations combines the advantages of the well known line charts and the pixel visualization, which provides identification of every single minute and enables recognition of recurring patterns. The overview also provides an interactive port activity map to identify the most active ports.

The *intrusion detection view* links IDS alerts or public warning lists with the full NetFlow records and displays the textual data in a colored table. By calculating some statistics concerning the influence of the attacking hosts on the whole network, the analyst gets a more realistic view of the relevance of the alerts. For further investigation of a number of hosts, it is possible to select the attackers and to visualize their connections to explore their influence. Besides the integration of external IDS alerts and warning lists, this view also provides a template editor to define database queries, which can directly access the flow data. We included a variety of different predefined templates, such as grabbing all SSH traffic or other suspicious activities.

Within the *flow visualization* view, we map the monitored network to a TreeMap visualization in the center of the display and arrange the previously selected attackers at the borders. The TreeMap comprises all hosts related to the attacking hosts during the chosen timeframe, which can be defined in the project creation wizard. Flows between the attackers and the local hosts or prefixes are displayed through Splines, whose control points are the center points of the network prefixes of various levels and the attackers on the outside. The size of the TreeMap rectangles (weight), their background color, and the Spline width can be set to arbitrary attributes of the aggregated flow data, e.g., flow count, transferred packets, or bytes.

In the default configuration the Spline color correlates with the attacker's IP prefix, which better shows the behavior of attackers with similar prefixes supporting the analyst in gaining insight into the distribution of the attacking IP addresses. Alternatively, random colors can be chosen.

The position of the attackers is calculated based on a *k-Medoid* clustering algorithm [11], which identifies all attackers and clusters them based on similar destination hosts. Therefore, it is possible to arrange hosts with similar victims close to each other to minimize overlaps. Another positive effect is the meaningful grouping of collaborating attackers in the same cluster.

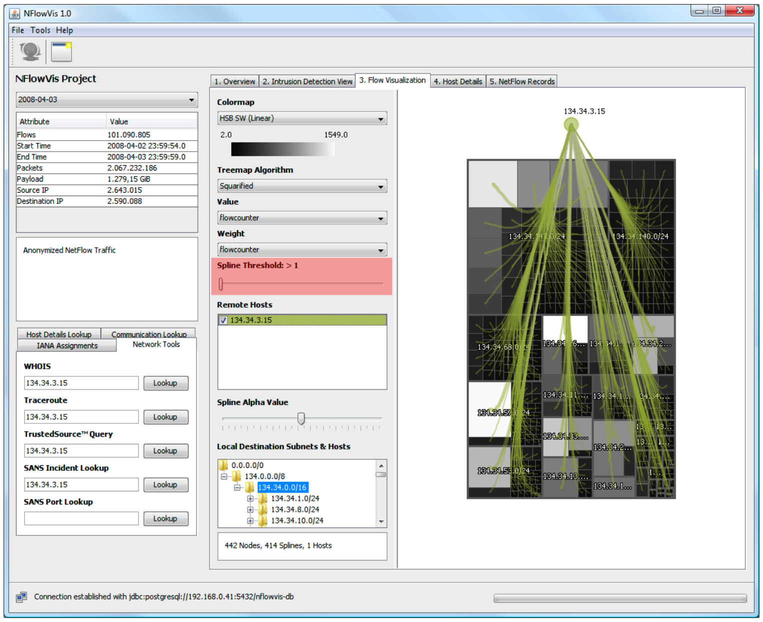
For further analysis of single hosts under attack, the analyst is able to use the *host view* detailing histograms, a port activity map, and an aggregated overview of all attackers related to the chosen host. Likewise, the original NetFlow records can be further analyzed by drilling-down and extracting the corresponding data in the *NetFlow records* view.

4 Case studies

In this section, we will demonstrate, how our *NFlowVis* tool can be applied to analyze potentially successful attacks, how massive distributed SSH attacks can be displayed, and how the tool can facilitate deeper understanding of service usage within the administrated IP network.

It is widely known that brute-force and dictionary SSH attacks are on the rise, as documented by huge numbers of explicit scans on port 22. The network security officer is primarily interested in those hosts, which do not have any prevention systems and do not block or throttle incoming login attacks. Since the attacker is able to make unlimited login attempts on such servers, the probability of successful logins is drastically increased. Besides, the attacker might automatically check the host's functionality after a successful login. By increasing a threshold slider in the visualization view, it is possible to smoothly hide all splines not exceeding that threshold (see Fig. 2) in order to identify hosts with high traffic to the attacker.

It is also possible to identify previously unknown attack scenarios. We were not aware of a massive distributed SSH attack in May 2008 before conducting a visual analysis of that day. The slow and low-volume attack pattern successfully avoided a) detection by intrusion prevention systems since it did not exceed common threshold limits and b) blocking of attacker IPs on the target machines. Using *NFlowVis*, we were able to identify a massive distributed SSH attack originating from a Botnet as shown in Fig. 3. The SSH connections originated from several hundred hosts, lasted over two days, and targeted about 50 specific university servers summing up to about 20 000 connections.



NFlowVis Project

2008-04-03

Attribute	Value
Flows	101.090.805
Start Time	2008-04-02 23:59:54.0
End Time	2008-04-03 23:59:59.0
Packets	2.067.232.186
Payload	1.279,15 GiB
Source IP	2.643.015
Destination IP	2.590.088

Anonymized NetFlow Traffic

WHOIS: 134.34.3.15

Traceroute: 134.34.3.15

TrustedSource™ Query: 134.34.3.15

SANS Incident Lookup: 134.34.3.15

SANS Port Lookup:

Colormap: HSB SW (Linear)

2.0 1549.0

Treemap Algorithm: Squarified

Value: flowcounter

Weight: flowcounter

Spline Threshold: > 1

Remote Hosts

☑ 134.34.3.15

Spline Alpha Value

Local Destination Subnets & Hosts

```
0.0.0.0/0
⊟ 134.0.0.0/8
  ⊟ 134.34.0.0/16
    ⊞ 134.34.1.0/24
    ⊞ 134.34.8.0/24
    ⊞ 134.34.10.0/24
```

442 Nodes, 414 Splines, 1 Hosts

Connection established with jdbc:postgresql://192.168.0.41:5432/nflowvis-db

Fig. 2. Identification of possibly compromised hosts using threshold adjustment (red)

Besides the analysis of attacks, *NFlowVis* can also be used to gain insight into legitimate network traffic, such as understanding of normal service usage or identification of abnormal network behavior. Fig. 4 details such a usage scenario in which the security officer visualizes flows to time servers. In this case, the IP addresses of the external time servers can reveal valuable information, for example, the distribution of operating systems on the machines within the local network since Macs have a tendency to connect to Apple's time servers and Windows machines preferably connect to Microsoft's time servers. Similar analysis can be conducted with IRC or DNS services, which might even be more relevant for network security.

5 Discussion

In contrast to previous approaches to visualizing traffic between internal and external network hosts, the combination of advanced visualization techniques with a clustering algorithm provides a scalable overview of the flows as demonstrated in Fig. 4. In particular, using the Hierarchical Edge Bundles [9] after clustering the external hosts based on the common internal connection hosts allowed us to identify distributed attacks and insightful traffic patterns as demonstrated in the previous section. In addition to that, the home-centric TreeMap visualization of the network is visually scalable due to the possibility to apply it at different granularity levels of the prefixes.

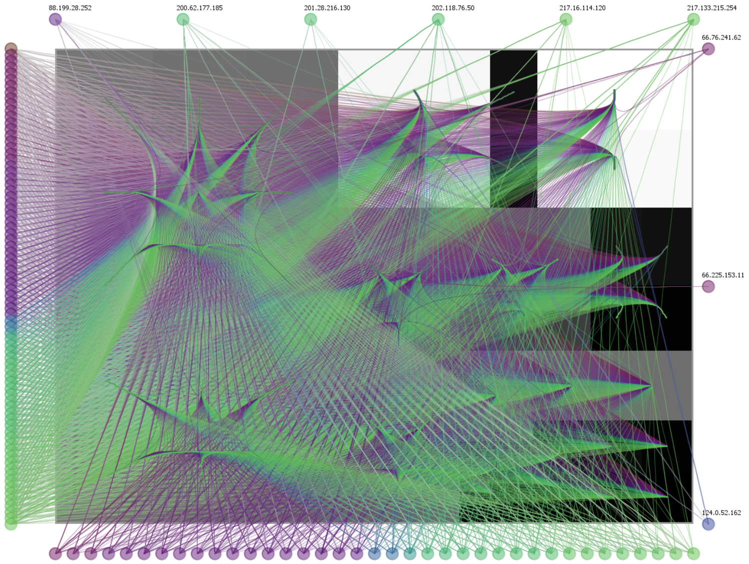


Fig. 3. Massive distributed SSH attack conducted by a Botnet with 120 Zombie computers against hosts of the university network on May 11, 2008.

6 Conclusions

In the scope of this paper, we presented the *NFlowVis* system to analyze intrusion detection and flow data. The user interface of the system follows a drill-down metaphor, guiding the analyst from an abstract overview of the overall network activity to aggregated views of IDS data and thorough analysis of attackers, their network traffic, and the victim hosts. In particular, this paper focused on a flow visualization technique combining a TreeMap visualization, a clustering algorithm, and Hierarchical Edge Bundles to group flows in a meaningful way. Three small case studies demonstrated the tool's applicability for exploring potentially successful attacks, for detection of slow and low-volume distributed attacks, and for analysis of service usage within our network.

In the future, we plan to extend our visualization technique to consider temporal aspects of attacks in more details using interactive specification of time intervals or a small multiples visualization.

Acknowledgment

This work has been funded by the BW-FIT research project "Information at your Fingertips: Interactive Visualization for Gigapixel Displays". We thank the anonymous reviewers of the VizSec 2008 for their valuable comments.

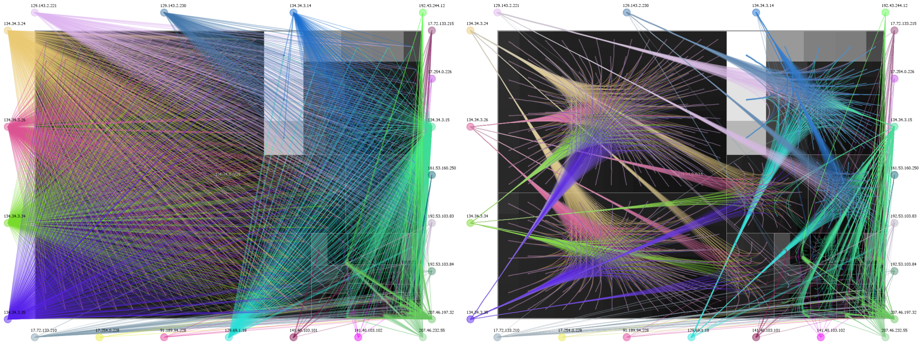


Fig. 4. Connections of local hosts to time servers visualized with straight lines (left) and Hierarchical Edge Bundles (right).

References

1. McPherson, D., Labovitz, C., Hollyman, M.: Worldwide infrastructure security report, Volume III. Technical report, Arbor Networks (September 2007)
2. NfSen - Netflow Sensor: A graphical web based front end for the nfdump netflow tools (2007) `http://nfsen.sourceforge.net/`.
3. Oslebo, A.: Stager A Web Based Application for Presenting Network Statistics. In: Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP. (2006) 1–15
4. Lakkaraju, K., Bearavolu, R., Slagell, A., Yurcik, W., North, S.: Closing-the-Loop in NVisionIP: Integrating Discovery and Search in Security Visualizations. In: Visualization for Computer Security, IEEE Workshops on. (26 Oct. 2005) 9–9
5. Goodall, J.R., Lutters, W.G., Rheingans, P., Komlodi, A.: Preserving the Big Picture: Visual Network Traffic Analysis with TNV. In: VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security, Washington, DC, USA, IEEE Computer Society (2005)
6. Ball, R., Fink, G., North, C.: Home-centric visualization of network traffic for security administration. Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security (2004) 55–64
7. Shneiderman, B.: Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph. **11**(1) (1992) 92–99
8. Mansmann, F., Keim, D.A., North, S.C., Rexroad, B., Sheleheda, D.: Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats. IEEE Transactions on Visualization and Computer Graphics **13**(6) (2007) 1105–1112
9. Holten, D.: Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. IEEE Trans. Vis. Comput. Graph. **12**(5) (2006) 741–748
10. Mansmann, F., Fischer, F., Keim, D., North, S.: Visualizing large-scale IP traffic flows. In: Proceedings of 12th International Workshop Vision, Modeling, and Visualization. (2007)
11. Kaufman, L., Rousseeuw, P.: Finding groups in data. An introduction to cluster analysis. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley (1990)