

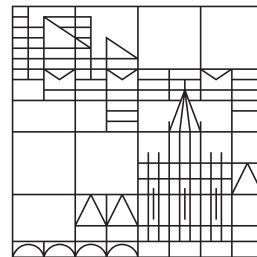
# Visual Analytics for Situational Awareness in Cyber Security

Dissertation zur Erlangung des akademischen Grades  
eines Doktors der Naturwissenschaften

vorgelegt von  
Fabian Fischer

an der

Universität  
Konstanz



Mathematisch-Naturwissenschaftliche Sektion  
Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 21. April 2016

1. Referent: Prof. Dr. Daniel. A. Keim
2. Referent: Jun.-Prof. Dr. Bela Gipp



# Abstract

More than ever, we rely on computer systems and the availability of computer networks. It is crucial to have a high standard of security in this modern world. Fully-automated systems to identify threats on the Internet are not enough to provide awareness of the actual situation of complex computer networks. Especially advanced persistent threats stay undetected for too long. Providing interactive visual interfaces in combination with analytical methods, help analysts and system administrators to get a better impression of possible symptoms, suspicious behavior, and understand complex dependencies to enhance cyber security. To achieve this goal, we implement and evaluate novel visual analytics systems to facilitate exploration of network activity, analysis of network threats, and correlation of heterogeneous data streams.

This thesis starts with an extensive literature review focusing on visualization systems supporting situational assessment in cyber security and identifies various research gaps. Afterwards, we focus on monitoring of network activity and introduce *VACS*, which is a web-based visual analytics suite for cyber security. This thesis also introduces a system for time-series analysis with integrated analytical methods to enhance visual correlation for port activity monitoring. Because of limitations of existing approaches to analyze temporal network data in a given hierarchical context, we also propose a novel visualization technique, called *ClockMap*. To assess this scalable approach, which is a unique combination of circular temporal glyphs and radial treemaps, we report the results of various evaluations. In particular, we actively participate in international challenges and successfully compete with other approaches and validate our findings based on ground truth data.

We also address the analysis of various specific cyber security threats. This thesis, therefore, proposes a novel visual analytics tool, called *VisTracer* to help network analysts to investigate BGP prefix hijackings and routing anomalies, which pose a severe threat to the underlying network infrastructure of the Internet. To make use of visual analytics to understand malware behavior, we contribute a taxonomy of visualization systems for malware analysis and reveal future research directions in this emerging field. Gaining situational awareness on a larger scale helps to understand the modus operandi of cyber attackers. We support this use case and integrate various alternative visualizations into *VACS* to facilitate attack attribution on multi-dimensional clusters. Furthermore, a field experiment with security experts is conducted to evaluate the novel combination of threat intelligence algorithms with interactive visual exploration.

The literature review shows that most of the visual analytics techniques in cyber security do not explicitly focus on dynamic real-time characteristics. However, concerning situational awareness, such capabilities are crucial. To emphasize the importance and foster more research in this direction, we propose a novel and scalable analysis infrastructure, integrated to *VACS*, for heterogeneous data streams. We specifically introduce, *NStreamAware*, which is a stream analysis system based on *Apache Spark*, and contribute a novel visualization technique, called *NVisAware*, to present aggregated data slices using various embedded visualization widgets to reduce the cognitive load of analysts. Moreover, visual feature selection techniques are applied to provide meaningful summaries of those slices. Eventually, we successfully evaluate the system using a network security case study and assess the general applicability in the context of situational awareness through active participation in an international competition.



# German Abstract

## —Zusammenfassung—

Mehr denn je sind wir heutzutage auf Computersysteme und die Verfügbarkeit von Computernetzwerken angewiesen. Deshalb sind hohe Sicherheitsstandards in unserer modernen Welt unabdingbar. Vollautomatische Systeme reichen allerdings nicht aus, um eine umfassende Einschätzung der aktuellen Bedrohungslage im Internet darzustellen und das Situationsbewusstsein für komplexe Computersysteme zu fördern. Insbesondere fortgeschrittene, andauernde Bedrohungen bleiben oftmals lange Zeit unentdeckt. Die Kombination von automatischen Analysemethoden und interaktiver visueller Benutzeroberflächen können dahingegen helfen, damit Analysten und Systemadministratoren einen besseren Blick für mögliche Auffälligkeiten erhalten und komplexe Zusammenhänge erfassen, um die IT-Sicherheit zu verbessern. Um dieses Ziel zu erreichen, implementieren und evaluieren wir im Rahmen dieser Arbeit innovative Visual Analytics Systeme, die dazu beitragen die Exploration von Netzwerkaktivität, Analyse von Netzwerkbedrohungen, und die Korrelation von heterogenen Datenströmen zu ermöglichen.

Diese Dissertation beginnt mit einer umfassenden Literaturrecherche und identifiziert verschiedene Forschungslücken. Anschließend legen wir den Schwerpunkt auf das Monitoring von Netzwerkaktivität und stellen *VACS* vor, welches eine webbasierte Visual Analytics Suite für IT-Sicherheit ist. Des Weiteren stellt die vorliegende Arbeit ein visuelles System mit integrierten analytischen Methoden zur Analyse von Zeitreihen vor, um die visuelle Korrelation im Rahmen des Monitorings von Port-Aktivität zu verbessern. Aufgrund Einschränkungen vorhandener Ansätze zeitliche Netzwerkdaten im jeweiligen hierarchischen Kontext zu analysieren, führen wir eine neuartige Visualisierungstechnik, *ClockMap*, ein. Um diesen skalierbaren Ansatz zu beurteilen, der auf einer Kombination von zirkulären Glyphen und radialen Treemaps basiert, beschreiben wir die Ergebnisse mehrerer Experimente. Im Besonderen nutzen wir die vorgestellte Technik, um diese durch aktive Teilnahme an internationalen Wettkämpfen zu vergleichen und die gewonnen Erkenntnisse zu verifizieren.

Im weiteren Verlauf dieser Arbeit betrachten wir weitere visuelle Methoden, um die Analyse verschiedener konkreter Bedrohungen der IT-Sicherheit zu unterstützen. Wir stellen das Visual Analytics Tool *VisTracer* vor, um Netzwerkanalysten zu helfen, sogenannte BGP-Prefix-Hijackings und Anomalien des Routings zu untersuchen, da diese eine folgenschwere Bedrohung für die grundlegende Netzwerkinfrastruktur darstellen. Um die Analyse von Schadsoftware zu verbessern, stellen wir eine Taxonomie für Visualisierungssysteme zur Malware-Analyse vor und zeigen weitere Forschungsperspektiven auf. Des Weiteren ist auch die Analyse auf globaler Ebene wichtig, um typische Vorgehensweisen von Angreifern zu ergründen. Um solche Anwendungen zu unterstützen, binden wir verschiedene Visualisierungen in *VACS* ein, um dadurch mehrdimensionale Cluster zu explorieren und die Zuordnung von Angriffen zu ermöglichen. Zudem führen wir mit IT-Sicherheitsexperten ein Feldversuch durch, um diese neuartige Kombination von Threat-Intelligence-Algorithmen und interaktiver visueller Exploration zu evaluieren.

Die Literaturrecherche zeigt, dass die meisten Visual Analytics Methoden im Bereich der IT-Sicherheit die besonderen Charakteristika von dynamischen Echtzeitdaten nicht berücksichtigen. Zur Verbesserung des Situationsbewusstseins sind diese allerdings entscheidend. Um dies zu verdeutlichen und hierbei einen Forschungsbeitrag zu leisten,

stellen wir ein neuartiges und skalierbares Analysesystem für heterogene Datenströme vor. Hierzu entwickeln wir *NStreamAware*, ein System basierend auf *Apache Spark*, und beschreiben eine Visualisierungstechnik mit dem Namen *NVisAware*, um aggregierte Teilstücke des Datenstroms mithilfe verschiedener Visualisierungs-Widgets darzustellen, um die kognitive Belastung der Analysten zu verringern. Des Weiteren setzen wir verschiedene Methoden der visuellen Featureauswahl ein, um sinnvolle Zusammenfassungen der Teilstücke zu berechnen. Im Anschluss evaluieren wir auch dieses System mithilfe realistischer Fallstudien und demonstrieren die generische Anwendbarkeit durch die aktive Teilnahme an einem internationalen Wettkampf.

# Acknowledgments

First of all, I would like to thank my advisor Prof. Dr. Daniel A. Keim for the great opportunity to be employed in his group and all the support over the last years. He was the one who initially motivated me to come to the University of Konstanz and shared his passion for research and visualization. I also would like to thank my second advisor, Jun.-Prof. Dr. Bela Gipp, for valuable and encouraging feedback. I would like to thank all my colleagues and collaborators, who are too many to list all of them. Their names can be seen in the various publications mentioned in the different sections of this dissertation.

However, I especially want to thank my dear colleague Johannes Fuchs. I really enjoyed the great and productive time, working and sharing the office with you. The time at the university would not have been nearly as much fun without you – keep on smiling! Furthermore, I thank Florian Mansmann, who involved me early on in paper writing and gave great guidance in getting started with research. I also thank Juri Buchmüller and Florian Stoffel for a great time, their reliability, and all the work we did together, especially in keeping the whole computer infrastructure of our group up and running. I also thank our support students (Udo Schlegel, Eren Cakmak) for their great work. Special thanks go to Matthew Sharinghousen, who took over my role as a system administrator, and is doing an excellent job! I also thank Martin Falk for letting me use and adapt his L<sup>A</sup>T<sub>E</sub>X template.

I want to thank all the former colleagues, who were part of the VIS-SENSE project, for the exceptional collaboration. My special thanks go to James Twellmeyer, Olivier Thonnard, and Pierre-Antoine Vervier.

I'm also thankful for funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257495, "*Visual Analytic Representation of Large Datasets for Enhancing Network Security*" (VIS-SENSE), which made it possible to work and do research with leading security experts.

Furthermore, I want to express my deepest gratitude to my parents, my whole family, and all my friends for their support. It is good to know that you are around – no matter what. However, most of all I want to thank my beloved wife, Judith. She is the best companion I can imagine – and I'm thankful for all her unconditional love and support that can hardly be expressed in words. Ultimately, I thank God for his grace and mercy, because he is "*before all things, and in him all things hold together*" (Colossians 1:17).

**Fabian Fischer**  
University of Konstanz  
January 2016





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	2
1.2	Research Goal . . . . .	3
1.3	Thesis Structure . . . . .	3
1.4	Publications . . . . .	4
<b>2</b>	<b>Visual Analytics for Situational Awareness</b>	<b>9</b>
2.1	Literature Review of Related Work . . . . .	10
2.1.1	Related Surveys . . . . .	10
2.1.2	Methodology . . . . .	12
2.1.3	Categorization and Taxonomy . . . . .	13
2.2	Observations and Research Gaps . . . . .	34
2.3	Research Objectives . . . . .	36
<b>3</b>	<b>Visual Analytics for Network Activity</b>	<b>39</b>
3.1	Visual Overview for Internal and External Monitoring . . . . .	40
3.1.1	VACS – Visual Analytics Suite for Cyber Security . . . . .	42
3.1.2	Evaluation using VAST Challenge 2013 . . . . .	48
3.1.3	Conclusions and Limitations . . . . .	57
3.2	Visual Correlation for Port Activity Monitoring . . . . .	58
3.2.1	IAS-Explorer – Visual Analytics for Port Activity Correlation . . . . .	61
3.2.2	Evaluation using Port Correlation Case Study . . . . .	65
3.2.3	Conclusions and Limitations . . . . .	68
3.3	Visual Exploration for Host and Server Monitoring . . . . .	69
3.3.1	ClockMap – Visualization Technique for Host Monitoring . . . . .	71
3.3.2	Evaluation of Alternative Glyph Designs . . . . .	77
3.3.3	Evaluation of ClockMap’s Design Principles . . . . .	80
3.3.4	Evaluation using VAST Challenge 2012 . . . . .	82
3.3.5	Evaluation using VAST Challenge 2013 . . . . .	90
3.4	Conclusions . . . . .	100
<b>4</b>	<b>Visual Analytics for Network Threats</b>	<b>103</b>
4.1	Visual Overview for Attack Patterns . . . . .	104
4.1.1	Usage of Temporal MDS Plots for Attack Patterns . . . . .	106
4.1.2	Evaluation using Network Security Case Study . . . . .	107
4.1.3	Conclusions and Limitations . . . . .	109
4.2	Visual Correlation for Routing Anomalies . . . . .	110
4.2.1	VisTracer – Visual Analytics for BGP Prefix Hijacking . . . . .	114
4.2.2	Evaluation using Case Studies . . . . .	120
4.2.3	Conclusions and Limitations . . . . .	126
4.3	Visual Analysis for Malware Behavior . . . . .	127
4.3.1	Taxonomy of Visualization Systems for Malware Analysis . . . . .	128
4.3.2	Conclusions and Limitations . . . . .	133

4.4	Visual Exploration for Attack Attribution . . . . .	135
4.4.1	Data Analytics for Threat Intelligence . . . . .	137
4.4.2	Integrated Visualizations for MDC Exploration . . . . .	137
4.4.3	Evaluation using Field Experiment . . . . .	138
4.4.4	Conclusions and Limitations . . . . .	142
4.5	Conclusions . . . . .	143
<b>5</b>	<b>Visual Analytics for Network Streams</b>	<b>145</b>
5.1	Visual Overview for Stream Monitoring . . . . .	148
5.1.1	Usage of Dynamic Visualizations for Stream Monitoring . . . . .	149
5.1.2	Conclusions and Limitations . . . . .	149
5.2	Visual Correlation for Heterogeneous Data Streams . . . . .	150
5.2.1	NStreamAware – Scalable Analytics for Data Streams . . . . .	150
5.2.2	Conclusions and Limitations . . . . .	152
5.3	Visual Exploration for Sliding Windows . . . . .	153
5.3.1	NVisAware – Visualization Technique for Sliding Slices . . . . .	153
5.3.2	Evaluation using Network Security Case Study . . . . .	157
5.3.3	Evaluation using VAST Challenge 2014 . . . . .	160
5.4	Limitations and Conclusions . . . . .	163
<b>6</b>	<b>Conclusions and Future Research Directions</b>	<b>165</b>
6.1	Summary . . . . .	165
6.2	Contributions . . . . .	167
6.3	Future Perspectives . . . . .	169
	<b>List of Figures</b>	<b>173</b>
	<b>List of Tables</b>	<b>181</b>
	<b>Bibliography</b>	<b>183</b>

*There is no reason for any individual  
to have a computer in his home.*

— Ken Olsen, DEC (1977)



## Introduction

### Contents

---

<b>1.1</b>	<b>Background</b>	<b>2</b>
<b>1.2</b>	<b>Research Goal</b>	<b>3</b>
<b>1.3</b>	<b>Thesis Structure</b>	<b>3</b>
<b>1.4</b>	<b>Publications</b>	<b>4</b>

---

TODAY’S WORLD heavily depends on mobile devices, embedded systems, computers, servers, networks, and the Internet. Recent developments and news reports show, that such systems are constantly under attack. Not only, less secured end users, but also highly secure computer networks like federal agencies have been successfully infiltrated in the past [33, p. 26].

Common cyber security threats often involve advanced persistent threats (APT), distributed denial-of-service (DDoS) attacks, cross-platform malware (CPM), metamorphic and polymorphic malware, phishing, BGP hijacks, cyber espionage, data breaches, vulnerabilities, malicious web sites, social media scam, credit card fraud, identity theft, and more. *“If there is one thing that can be said about the threat landscape, and Internet security as a whole, it is that the only constant is change”* [231] as stated in Symantec’s *2015 Internet Security Threat Report* [231]. The wide variety and the increase of sophisticated, ever-changing, attacks highlight the importance of research in the area of cyber security. One important objective is also to teach users, because in recent years most of the successful attacks to highly secured networks often started with social engineering and a weak link, which is often a user opening a seemingly legitimate e-mail attachment.

A major incident of such advanced persistent threat (APT) became public in May 2015, in which the internal computer network of the German parliament (Bundestag), called *Parlakom*, was successfully compromised by unknown attackers. And as common for such attacks, it started with a simple, but carefully crafted, spear phishing: An e-mail which looked like a legitimate letter from *un.org* [117] was sent to specific members

of parliament with links to a malicious website. After visiting the website, malicious code was installed, which further infiltrated the computer network. In this case, the attackers could stay undetected for months and most likely could exfiltrate sensitive data. Eventually, officials decided to take down the computer network to investigate the incident and deploy more measures and systems to strengthen the network security.

While there are automated detection systems to block known malware samples, using anti-virus appliances, it is hard to detect samples, which are specifically built to target a particular user or organization. Having said that, it becomes obvious that there will always be a way for criminals to find an attack vector to get into a computer network. Therefore, it is impossible to prevent every (targeted) attack automatically. But we still need technology and ways, so that successful attackers cannot stay undetected for too long. Therefore, this thesis contributes various techniques to help analysts to detect and discover symptoms or anomalies in a timely manner and better understand the overall modus operandi of attack campaigns.

## 1.1 Background

There are many security-related policies, best practices, and regulations available to provide guidelines for secure computer systems and how to get certified according to such standards. Müller [174] provides an extensive overview about the most important standards including the IT baseline protection (IT-Grundschutz) as defined by the German Federal Office for Information Security, which is compliant to the ISO/IEC 27000 series of information security standards. Detailed standards (e.g., PCI DSS) are proposed by the payment card industry to ensure secure processing of credit card data on computer systems. While some of the work discussed in this thesis could be more precisely described as work in the field of operational “computer network security”, the overall scope of this thesis is broader, because it includes an extensive review and work in the area of (forensic) malware analysis and strategic threat analysis, so the usage of the term “cyber security” is more appropriate. The International Telecommunication Union (ITU) also suggests various recommendations for cyber security, and defines cyber security in ITU-T X.1205 [128] quite general as “*the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user’s assets*” [128]. Eventually, cyber security “*strives to ensure the attainment and maintenance of the security properties of the organization and user’s assets against relevant security risks in the cyber environment*” [128]. However, because of the complexity of current and future attacks, we need to keep the human analyst in the loop to also enhance situational awareness (SA) for decision makers. We strongly believe, that visual analytics is an approach specifically helpful in this domain, because it “*is the science of analytical reasoning facilitated by interactive visual interfaces*” [247] and combines the strengths of automated processing power of modern computer systems with expert knowledge and intuition of human analysts. Thomas et al. [247] also state that the “*analysis of overwhelming amounts of disparate, conflicting, and dynamic information is central to identifying and preventing emerging threats, (...) and responding in the event of an attack or other disaster*”. The human is inevitable, because of the “*rapidly changing situations to both detect the expected and discover the unexpected*” [247]. Also in the context of cyber security, the human is quite good in judging unexpected events, and it has been shown that visualization helps the analyst to acquire a higher number and even more accurate insights [100].

Not only identifying attacks, but also providing a better understanding of the current network situation is crucial for cyber security. Attacks might result in anomalies and side-effects which can be identified through outages of specific services leading to obvious changes in network traffic. Therefore, it is not only important to identify specific attacks and be aware of current alerts of intrusion detection systems, but also have awareness of the current operational network situation. Furthermore, such incidents need to be analyzed within their context, otherwise they are hard to interpret. This is the reason why fully automated systems, might not be appropriate for complex attacks, because such situations can only be interpreted within the context. The advantage of visual techniques is, that the analyst, for example, is able to quickly explore such anomalies with respect to the behavior of other hosts in the sub network. Therefore, we believe that the usage of visual analytics provides a promising direction to gain situational awareness to eventually enhance cyber security.

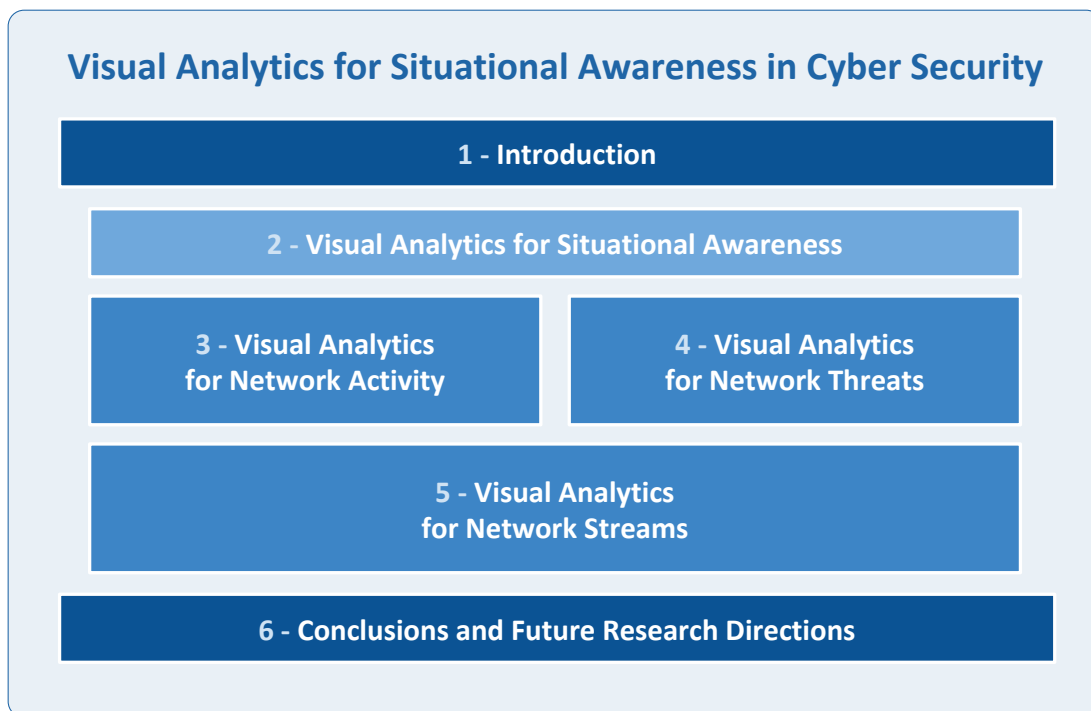
## 1.2 Research Goal

Because of the high relevance of research in cyber security and the observed limitations of *fully-automated* intrusion detection systems to discover unexpected, unknown, and complex anomalies, the following research goal was defined to better include humans' capabilities:

**Propose, implement, and evaluate interactive visualization systems to enhance situational awareness in cyber security through the scalable exploration of network activity, the analysis of network threats, and visual analytics support for the analysis of heterogeneous data streams by combining automated methods with scalable and interactive visualizations.**

## 1.3 Thesis Structure

To best address the general question on how to enhance situational awareness using visual analytics, this thesis is structured as follows as seen in Figure 1.1. Chapter 2 briefly defines situational awareness and discusses general aspects about cyber security with respect to visual analytics and presents state of the art in the field of visualizations to enhance situational awareness. In Chapter 3, we focus on the analysis of network activity that is often related to temporal network data. We present various techniques to visually analyze such time-series data in the context of situational awareness and introduce a novel visualization for hierarchical time-series data, which can also be applied to other domains. Chapter 4 focuses more on networks threats and show how visual analytics can be used to visually explore actual network threats. Specifically, we focus on a visual analytics system to analyze routing anomalies with respect to BGP hijacking events. Furthermore, we present a taxonomy for visualization systems for malware analysis and address an open research gap using alternative visualizations to analyze the general threat landscape for attack attribution. Chapter 5 introduces a scalable system, which applies visual analytics to heterogeneous data streams for situational awareness. Because evaluating complex security applications is challenging, we actively participate and compete in various international competitions as promising evaluation strategy for security applications and report on these results within the respective chapters. Chapter 6 concludes with a summary and suggests various future research perspectives.



▲ **Figure 1.1 — Overview of thesis structure.** After the introduction, Chapter 2 presents an extensive literature review in the field of visual analytics for cyber security with a focus on situational awareness. Chapter 3 focuses on visual analysis of network activity, while Chapter 4 focuses on network threats explicitly. Chapter 5 tackles the real-time challenge for situational awareness on heterogeneous data streams. Chapter 6 concludes the thesis and summarizes the contributions.

For better readability and to reflect the fact that many of the ideas were discussed and published together with other researchers, I decided to use mostly “we” instead of “I”. In the beginning of the various sections, I include footnotes to clearly highlight the individual contributions of the various authors.

## 1.4 Publications

To share the results of this thesis with the community in a timely manner, so other researchers are able to build upon this work, most parts of this thesis have been previously published in well known venues over the past years, which is common practice for computer science doctoral theses. Therefore, this thesis is based on the following publications.

### Surveys

- E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P.-A. Vervier. Visual Analytics for BGP Monitoring and Prefix Hijacking Identification. *IEEE Network*, 26(6):33–39, 2012. ISSN 0890-8044. doi:10.1109/MNET.2012.6375891 [25].

- M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner. A Survey of Visualization Systems for Malware Analysis. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARS*, Italy (Cagliari), 2015. The Eurographics Association. doi:10.2312/eurovisstar.20151114 [261].

### Applications / Design Studies

- F. Fischer, J. Fuchs, and F. Mansmann. ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data. In M. Meyer and T. Weinkauff, editors, *Proceedings of the Eurographics Conference on Visualization (EuroVis - Short Papers)*, pages 97–101, Vienna, Austria, 2012. The Eurographics Association. ISBN 978-3-905673-91-3. doi:10.2312/PE/EuroVisShort/EuroVisShort2012/097-101 [82].
- F. Fischer, J. Fuchs, P.-A. Vervier, F. Mansmann, and O. Thonnard. VisTracer: A Visual Analytics Tool to Investigate Routing Anomalies in Traceroutes. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 80–87, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379701 [84].
- F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. Visual Analytics zur Firewall-Konfiguration und Analyse von Netzwerkverkehr (in German). In B. f. S. i. d. Informationstechnik, editor, *Informationssicherheit stärken - Vertrauen in die Zukunft schaffen: Tagungsband zum 13. Deutschen IT-Sicherheitskongress (in German)*, pages 273–283. SecuMedia Verlag, 2013 [86].
- F. Stoffel, F. Fischer, and D. A. Keim. Finding Anomalies in Time-Series using Visual Correlation for Interactive Root Cause Analysis. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security, VizSec '13*, pages 65–72, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517966 [226].
- F. Fischer and D. A. Keim. NStreamAware: Real-Time Visual Analytics for Data Streams to Enhance Situational Awareness. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 65–72, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671495 [79].
- F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: Visual Analytics for Big Data in Large-Scale Computer Networks. *Information Visualization*, 14(1):51–61, 2015. ISSN 1473-8716, 1473-8724. doi:10.1177/1473871613488572 [90].
- D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):141–150, 2016. ISSN 1077-2626. doi:10.1109/TVCG.2015.2467553 [133].

### Evaluations

- J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*,

CHI '13, pages 3237–3246, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi:10.1145/2470654.2466443 [95].

- F. Fischer, J. Davey, J. Fuchs, O. Thonnard, J. Kohlhammer, and D. A. Keim. A Visual Analytics Field Experiment to Evaluate Alternative Visualizations for Cyber Security Applications. In M. Pohl and J. Roberts, editors, *Proc. EuroVA International Workshop on Visual Analytics*. The Eurographics Association, 2014. ISBN 978-3-905674-68-2. doi:10.2312/eurova.20141144 [88].

### Challenge Submissions

Additionally, we successfully participated in various challenges to evaluate our approaches with realistic scenarios and compete with international teams around the world. In the following a list of only those submissions, which directly contribute to this thesis.

- **VAST Challenge 2012** [52] – We participated in Mini-Challenge 1 (MC1) and Mini-Challenge 2 (MC2) and won an award for an “*outstanding comprehensive submission*” [52]:
  - F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: A Visual Situational Awareness Tool for Large-Scale Computer Networks (VAST Challenge 2012). In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 257–258, 2012. doi:10.1109/VAST.2012.6400528 [83].
- **VAST Challenge 2013** [269] – We participated in particular in MC2 and Mini-Challenge 3 (MC3) and received an honorable mention for an “*Interesting Visualization Technique*” [269] for MC2 and a honorable mention for an “*Intriguing Visualization*” [269] for MC3:
  - F. Fischer, D. Jäckle, D. Sacha, F. Stoffel, and D. A. Keim. Adaptive User-Aware Dashboard Design. In *VAST Challenge 2013 - Honorable Mention*, 2013 [87].
  - F. Fischer and D. A. Keim. VACS: Visual Analytics Suite for Cyber Security - Visual Exploration of Cyber Security Datasets. In *VAST Challenge 2013 - Honorable Mention*, 2013 [78].
- **VAST Challenge 2014** [270] – We participated in all mini-challenges. The ones relevant in the scope of this thesis are MC3 and the Grand Challenge (GC) combining all mini-challenges. For the GC, we received an honorable mention for an “*Effective Analytic Presentation*” [270] and also got an award for an “*Outstanding Comprehensive Mini-Challenge 3 Submission*” [270].
  - F. Fischer, F. Stoffel, S. Mittelstädt, T. Schreck, and D. A. Keim. Using Visual Analytics to Support Decision Making to Solve the Kronos Incident (VAST Challenge 2014). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 301–302, 2014. doi:10.1109/VAST.2014.7042537 [89].
  - F. Fischer and F. Stoffel. NStreamAware: Real-Time Visual Analytics for Data Streams (VAST Challenge 2014 MC3). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 373–374, 2014. doi:10.1109/VAST.2014.7042572 [80].



## Technical Reports

Parts of the research contributed to this thesis, was funded work done within the *VIS-SENSE*<sup>1</sup> project. Therefore, various parts of this thesis were previously made available online as technical deliverable reports, accessible on the *VIS-SENSE* project website<sup>2</sup>. In particular, I personally contributed to the following deliverable reports:

- D1.1 Analysis of Current Practices (M6)
- D3.1 Specification of the Network Analytics Algorithms (M9)
- D3.3 Attack Attribution Module (M24)
- D4.1 Visual Network Analysis Module (M24)
- D4.2 Visual Correlation Analysis Module (M24)
- D4.3 Visual Analysis System for Interactive Scalable Analysis (M24)
- D6.1 Threat Landscape Identification Scenario (M36)
- D6.2 BGP Analysis Scenario (M36)
- D6.3 VIS-SENSE Framework Evaluation (M38)

## Other Publications

In addition, there are a number of related projects I was involved in during my time as PhD student, that only indirectly contribute to the content of this thesis. More information on this work can be found in the following publications:

- C. Rohrdantz, D. Oelke, M. Krstajic, and F. Fischer. Real-Time Visualization of Streaming Text Data: Tasks and Challenges. In *Workshop on Interactive Visual Text Analytics for Decision-Making at the IEEE VisWeek 2011*, 2011 [201].
- E. Bertini, J. Buchmüller, F. Fischer, S. Huber, T. Lindemeier, F. Maaß, F. Mansmann, T. Ramm, M. Regenscheit, C. Rohrdantz, C. Scheible, T. Schreck, S. Sellien, F. Stoffel, M. Tautzenberger, M. Zieker, and D. A. Keim. Visual Analytics of Terrorist Activities Related to Epidemics. In *Proc. IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2011 - Grand Challenge Award)*, 2011. doi:10.1109/VAST.2011.6102498 [23].
- F. Mansmann, M. Krstajic, F. Fischer, and E. Bertini. StreamSqueeze: A Dynamic Stream Visualization for Monitoring of Event Data. In *Proceedings of Conference on Visualization and Data Analysis (VDA '12)*, volume 8294, pages 829404–829404–12, 2012. doi:10.1117/12.912372 [169].
- F. Mansmann, F. Fischer, and D. A. Keim. Dynamic Visual Analytics – Facing the Real-Time Challenge. In J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, editors, *Expanding the Frontiers of Visual Analytics and Visualization*, pages 69–80. Springer London, 2012. ISBN 978-1-4471-2803-8 978-1-4471-2804-5 [167].

<sup>1</sup> *VIS-SENSE* was a funded project from the European Commission’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257495, “*Visual Analytic Representation of Large Datasets for Enhancing Network Security*”.

<sup>2</sup> [www.vis-sense.eu](http://www.vis-sense.eu)

- M. Behrisch, J. Davey, F. Fischer, O. Thonnard, T. Schreck, D. Keim, and J. Kohlhammer. Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom. *Computer Graphics Forum*, 33(3):411–420, 2014. ISSN 1467-8659. doi:10.1111/cgf.12397 [20].
- J. Fuchs, P. Isenberg, A. Bezerianos, F. Fischer, and E. Bertini. The Influence of Contour on Similarity Perception of Star Glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2251–2260, 2014. ISSN 1077-2626. doi:10.1109/TVCG.2014.2346426 [96].
- J. Fuchs, R. Rädle, D. Sacha, F. Fischer, and A. Stoffel. Collaborative Data Analysis with Smart Tangible Devices. In *Proceedings of Conference on Visualization and Data Analysis (VDA '14)*, volume 9017, pages 90170C–90170C–15, 2014. doi:10.1117/12.2040011 [97].
- F. Stoffel and F. Fischer. Using a Knowledge Graph Data Structure to Analyze Text Documents (VAST Challenge 2014 MC1). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 331–332, 2014. doi:10.1109/VAST.2014.7042551 [225].
- M. El Assady, W. Jentner, M. Stein, F. Fischer, T. Schreck, and D. A. Keim. Predictive Visual Analytics – Approaches for Movie Ratings and Discussion of Open Research Challenges. In *Proceedings of the IEEE VIS 2014 Workshop Visualization for Predictive Analytics*, 2014 [64].
- D. Streeb, U. Schlegel, J. Buchmüller, F. Fischer, and D. A. Keim. Using visual analytics to analyze movement and action patterns. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 171–172, 2015. doi:10.1109/VAST.2015.7347665 [228].
- B. Schneider, C. Acevedo, J. Buchmüller, F. Fischer, and D. A. Keim. Visual analytics for inspecting the evolution of a graph over time: Pattern discovery in a communication network. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 169–170, 2015. doi:10.1109/VAST.2015.7347664 [207].
- E. Cakmak, A. Gartner, T. Hepp, J. Buchmüller, F. Fischer, and D. A. Keim. Applying visual analytics to explore and analyze movement data. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 127–128, 2015. doi:10.1109/VAST.2015.7347643 [36].
- J. Buchmüller, F. Fischer, D. Streeb, and D. A. Keim. Using visual analytics to provide situation awareness for movement and communication data. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–122, 2015. doi:10.1109/VAST.2015.7347640 [34].

*“Let us not look back in anger or  
forward in fear, but around in  
awareness.”*

— James Thurber



## Visual Analytics for Situational Awareness

### Contents

---

<b>2.1 Literature Review of Related Work</b> . . . . .	<b>10</b>
2.1.1 Related Surveys . . . . .	10
2.1.2 Methodology . . . . .	12
2.1.3 Categorization and Taxonomy . . . . .	13
<b>2.2 Observations and Research Gaps</b> . . . . .	<b>34</b>
<b>2.3 Research Objectives</b> . . . . .	<b>36</b>

---

SITUATIONAL awareness and cyber security has a strong need for visualization support to involve the analyst in complex data analysis tasks.

Colloquially speaking the definition of situational awareness (SA) can be summarized with the simple statement of *“knowing what’s going on so you can figure out what to do”* [3]. This is also relevant for an airplane pilot checking and monitoring the various instruments within the cockpit and observing potential threats outside the aircraft. So what does it mean to have “good” situational awareness? How can that be evaluated and measured? Obviously, this is quite challenging, because it refers to a mental state of the pilot. This general concept of situational awareness is not only relevant for aviation, but for many real-world scenarios. It is crucial in healthcare that a surgeon is aware of the current situation, a driver in a car needs to know what is going on around the car, and a decision maker needs to be aware of the whole circumstances to make the right decisions. This is not only true in the physical world, but also in the cyber world. A network operators needs to know what is going on, to understand the current situation, to protect the network, mitigate attacks, or to be aware of the risks of a potential malware threat.

Most theories about situational awareness have their roots in aviation and were primarily influenced by Mica Endsley (e.g., see [66, 67]). She conducted extensive research in SA and was Chief Scientist at United States Air Force and defined SA as *“the perception of the elements in the environment within a volume of time and*

space, the comprehension of their meaning, and the projection of their status in the near future” [67]. The concept of SA in the physical world was quickly adopted also to the cyber world (e.g., [235, 17]). In the scope of this thesis, we follow the intentions by Franke and Brynielsson [93] and “*think of situational awareness primarily as a mental state that can be reached to a varying degree*” [93] which relates to the various states defined by Endsley, which will be briefly discussed in Section 2.1.3.

### Situational Awareness / Assessment

Situational Awareness (SA) as a mental state can be referred to as a state of knowledge, which can be achieved using various techniques. In the world of cyber security, systems and tools exist to (visually) analyze, explore, and monitor the current situation leading to findings, insights, and eventually knowledge. The process to gain that knowledge can be “*referred to as **situation[al] assessment** or as the process of achieving, acquiring, or maintaining SA*” [67].

## 2.1 Literature Review of Related Work

The following sections give a broad overview of the current state of research with respect to visual analytics in the domain of cyber security. First, we present various related surveys, discuss their shortcomings and contribute a detailed and comprehensive state-of-the-art literature review for the research area. We introduce the various categories and results here, and make use of more detailed results, tables, and discussions within the respective chapters and sections throughout the dissertation.

### 2.1.1 Related Surveys

There are various surveys and literature reviews in the broader area of security visualization. In 2009, **Tamassia et al.** [238] review graph drawing techniques used in 16 computer security visualizations. They identify graph drawing techniques to support network monitoring, BGP analysis, access control, trust negotiation, and attack graphs. Because the authors limit their survey to graph techniques, the work cannot present a comprehensive view of security-related visualization systems. **Zhang et al.** [282] focus in 2012 on a particular data type – computer network logs. They provide a survey of security visualizations for this data source and distinguish between the distinctive visualization designs. They classify the proposed tools in text-based, parallel, hierarchical, three dimensional, and other forms of visual representations. While this is a good start, the view is quite limited to the visual analysis of computer network logs, which is only a particular data type interested for security analysts.

**Shiravi et al.** [216] fill this gap and present a comprehensive survey of visualizations systems for network security in 2012 and review 45 publications and facilitate an use case based approach to classify the reviewed tools. Shiravi et al. [216] also conclude that the “*process of achieving situational awareness is closely related to the capability of a system in conducting real time analysis. Security visualization systems, in their current state, are mostly suitable for offline forensics analysis.*” [216]. This statement highlights the need for research to provide real-time capabilities in cyber security visualizations as we propose in Chapter 5 for data streams. **Li et al.** [158] does not take a use case based approach, but focus on a limited selection of tools using network flow data as primary

data source. The authors survey state of the art for analysis methods and visualization approaches specifically for network flow analysis up to the year 2012. Because of the security critical impact of prefix hijacking and the lack of literature reviews in this field – even Shiravi et al. [216] only briefly mention a few tools – we published in 2012 a survey together with **Biersack et al.** [25] reviewing visual analytics tools for BGP monitoring. Here we review 9 tools, with respect to level of details, visualization techniques, features, and applicable use cases, which will be discussed in Chapter 4. **Harrison and Lu** [114] focus on a detailed review and comparison of few selected security visualizations for network data rather than on a complete literature review in 2012. However, they reveal strengths and weaknesses of the respective tools and propose future directions, especially emphasizing the need for more scalable solutions. **Tran Khanh Dang and Tran Tri Dang** [251] survey security visualization techniques for web information systems with a different point of view. They distinguish the proposed visualizations mainly in client- and server-side systems. Especially the client-side systems are not in the scope of most other surveys. The authors describe proactive and reactive approaches (e.g., intrusive and non-intrusive warnings), for example to present custom visualizations to the visitor of a website, to help to distinguish between phishing/spoofed and real websites using visual techniques. They also cover a limited number of server-side systems to visually explore network packet, flow, and application generated data.

In 2014, **Franke and Brynielsson** [93] specifically focus on cyber situational awareness and conduct a systematic review of literature. Their survey is quite broad and focuses also on publications which are not related to visualization. They focus for example on introductory literature on cyber situational awareness, SA in industrial control systems, SA in emergency management, SA architectures and algorithms, and on establishing nation-wide cyber situational awareness. They also focus in one section explicitly on visualization support for cyber SA and human-computer interaction. However, this overview is quite incomplete, because many publications not directly talking about situational awareness, do still provide visual exploration and monitoring techniques to support situational assessment to eventually contribute and enhance the mental state of SA. This shows that a more comprehensive literature review is needed to include even more visualization systems relevant for cyber SA.

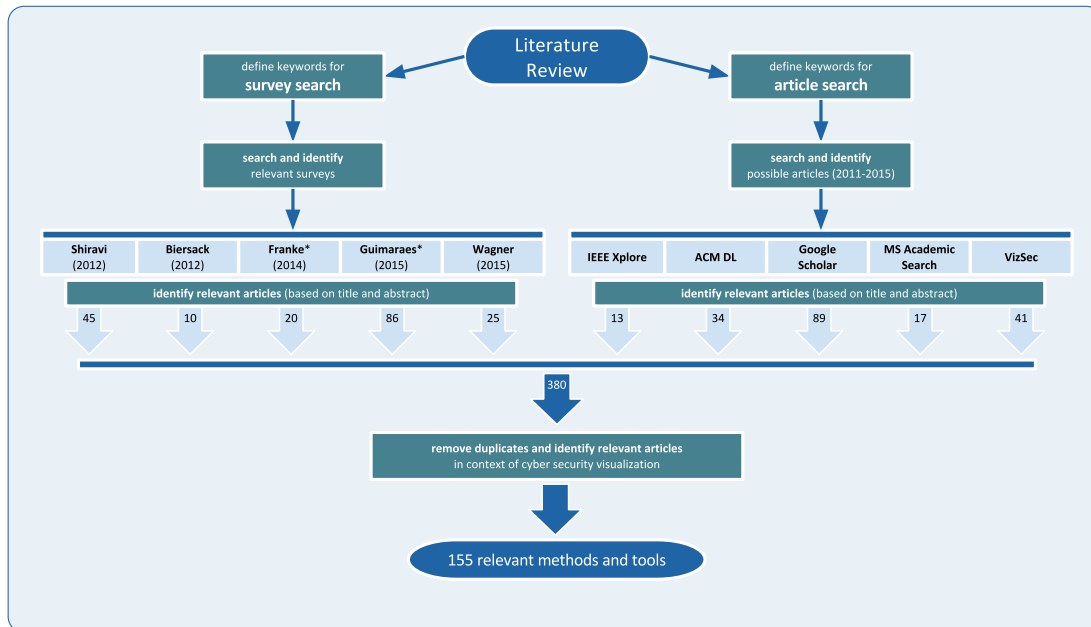
In 2015, **Guimaraes et al.** [108] present an extensive survey on information visualization for network and service management classifying “285 articles and papers from 1985 to 2013, according to an information visualization taxonomy, as well as a network and service management taxonomy” [108]. Because of their quite general topic-based taxonomy, the authors provide a nice historic overview starting from papers published in the 80’s until 2008, and describe relevant tools between 2009 to 2013 in a state-of-the-art report in more details. Their survey also reveals, that most of the relevant articles are published at the symposium of *Visualization for Cyber Security* (VizSec)<sup>1</sup>. However, the taxonomy is quite general and lacks a more detailed classification within the subtopic *IP networks*. **Staheli et al.** [223] provide a survey in 2014 of all visualization evaluations for cyber security published at VizSec in the last decade. The authors identify most common evaluation types for complex security applications and reveal trends and future directions.

Over the years, a noticeable trend could be identified, that there is an increasing body of research of visualization systems for malware analysis. To provide a state-of-the-art report about novel techniques in this field, we conducted an extensive literature review

---

<sup>1</sup> [www.vizsec.org](http://www.vizsec.org)

together with **Wagner et al.** [261] in 2015. In this work, we review 25 malware visualization systems and propose a malware visualization taxonomy to classify the systems into distinctive categories, which is also discussed in Chapter 4.



▲ **Figure 2.1 — Methodology of literature review.** The literature review is based on a combination of papers identified within existing surveys and keyword search in various digital libraries to include recent state of the art.

### 2.1.2 Methodology

The survey presented here, is the most comprehensive literature review in the field of visualization and visual analytics with focus on cyber security and situational awareness. It incorporates all the publications reviewed in the well-structured survey by Shiravi et al. [216]. However, we also extend the scope to the threat landscape and malware analysis, which Shiravi et al. [216] did not include in their review. Additionally, we extend the literature research to incorporate the most recent publications in the field until 2015 and the identified articles within the aforementioned surveys. The initial starting point for related research were papers published at the premier forum for *Visualization for Cyber Security* (VizSec)<sup>2</sup>. This venues “brings together researchers and practitioners from academia, government, and industry to address the needs of the cybersecurity community through new and insightful visualization and analysis techniques” [259], for which we also contribute a web-based overview<sup>3</sup> of all papers published at VizSec, which is linked from the official conference website. Additionally, we made use of a number of common digital libraries (IEEE Xplore, ACM digital library, Google Scholar) and searched for relevant keywords and especially focused on recent publications from 2012 to 2015, to

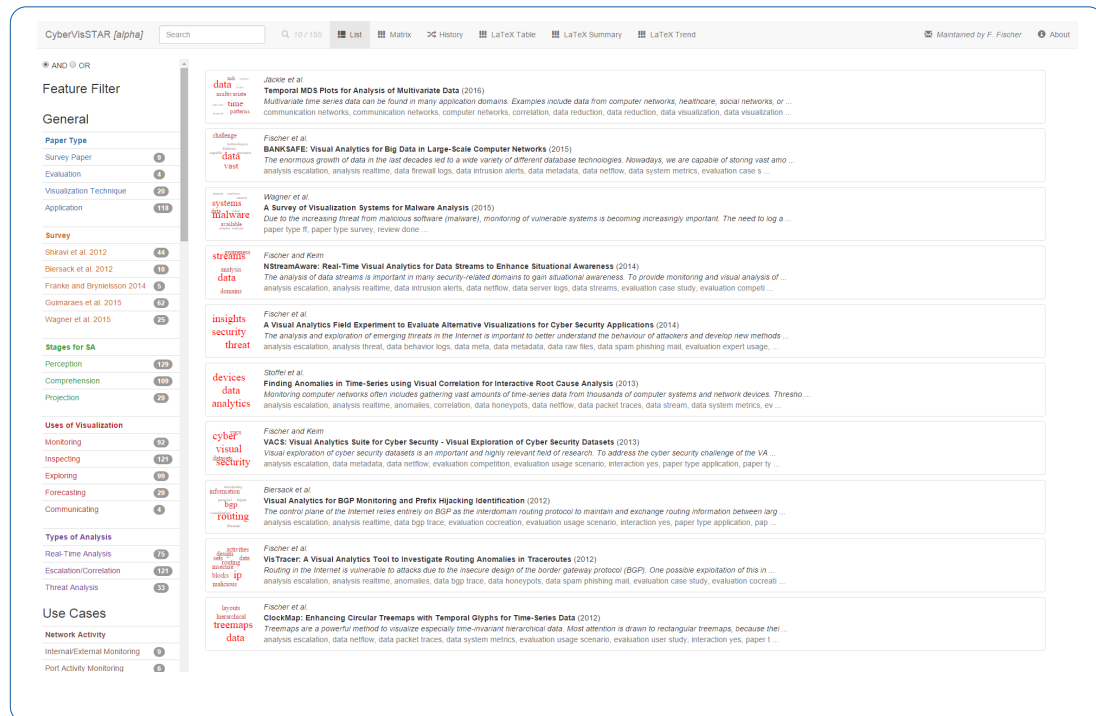
<sup>2</sup> [www.vizsec.org](http://www.vizsec.org)

<sup>3</sup> [vizsec.dbvis.de](http://vizsec.dbvis.de)

include state-of-the-art work, not discussed in the aforementioned surveys. Figure 2.1 presents the general workflow we used to conduct the comprehensive literature review for visualization approaches and tools to enhance situational awareness in cyber security.

### 2.1.3 Categorization and Taxonomy

We eventually identified a total of 155 academic articles and classified them according to various taxonomies, which are briefly introduced in the following sections. Parts of this survey were previously published in various publications [25, 261] and were partly made publicly available as web applications<sup>4</sup> to share the results with the community. An interactive web-based visual library summarizing the overall literature review for cyber security visualizations of this thesis can also be found online<sup>5</sup>. This web-based exploration tool, as seen in Figure 2.2, helps to make interesting observations, to identify trends, and reveal research gaps for the current state of the art.



▲ **Figure 2.2** — A survey of visualization systems for cyber security. An extensive web-based literature review of visualization systems for cyber security.

### Paper Types

The general paper type is just a basic categorization into (i) *Survey Paper*, (ii) *Evaluation Paper*, (iii) *Visualization Technique*, and (iv) *Application Paper* as presented in Table 2.1. We treat visualization techniques and application papers mostly as disjoint from each

<sup>4</sup> vizsec.dbvis.de, malware.dbvis.de

<sup>5</sup> cybervis.dbvis.de

other. While there are various application papers, which also present novel visualization techniques, the main focus of these papers is generally not the visualization techniques but the application focus instead. Therefore, we used the visualization technique category only, when the technique was the primary focus and could be applied to various other data sources in the field of cyber security. Articles in the category of evaluation papers focus on the comparison of various visualization techniques for cyber security and evaluate them; however in most cases they do not primarily focus on a new application or technique.

▼ **Table 2.1 — State-of-the-art overview according to paper type.** The table gives an overview of the general paper types included in the overall literature review.

Category	Methods <sup>6</sup>
Survey Paper	[282] [216] [ <b>25</b> ] [114] [158] [93] [153] [108] [ <b>261</b> ]
Evaluation	[243] [100] [8] [ <b>88</b> ]
Visualization Technique	[69] [68] [14] [143] [161] [160] [70] [165] [22] [81] [131] [189] [276] [71] [ <b>82</b> ] [169] [285] [290] [286] [ <b>133</b> ]
Paper Type Application	[99] [180] [237] [242] [241] [209] [18] [142] [149] [173] [245] [277] [278] [244] [49] [145] [271] [102] [155] [198] [45] [55] [91] [148] [181] [199] [246] [50] [273] [182] [164] [27] [163] [172] [185] [191] [193] [239] [51] [94] [166] [195] [252] [240] [101] [46] [119] [21] [24] [47] [159] [215] [260] [279] [28] [73] [105] [140] [176] [203] [53] [196] [130] [9] [85] [ <b>84</b> ] [106] [115] [125] [157] [168] [206] [211] [253] [275] [30] [186] [ <b>25</b> ] [221] [7] [ <b>78</b> ] [98] [109] [112] [113] [121] [156] [187] [188] [ <b>226</b> ] [289] [272] [134] [63] [179] [41] [ <b>79</b> ] [ <b>88</b> ] [92] [104] [122] [162] [213] [224] [263] [111] [110] [287] [212] [144] [151] [281] [42] [ <b>90</b> ] [264] [183] [12] [37]

## Survey

As discussed in Section 2.1.1, we include publications also reviewed in other literature reviews and surveys. To highlight these relations, we use this category to show, which approaches were reviewed by the most important existing surveys. This helps to quickly identify those approaches which were not part of any previous literature review. We include the surveys by Shiravi et al. [216], Biersack et al. [25], Franke and Brynielsson [93], Guimaraes et al. [108], and Wagner et al. [261]. Table 2.2 summarizes the results with respect to the aforementioned surveys.

<sup>6</sup> References which are emphasized using bold font actually refer to methods contributed by this dissertation, but have been previously published. However, for consistency reasons I decided to include them in this overview.

<sup>7</sup> Please note that some of the individual surveys actually review more publications than listed here. I list only those papers which are also within the dissertation's scope of cyber security visualizations.



▼ **Table 2.2** — **State-of-the-art overview of related surveys.** Categorization of papers reviewed by various existing surveys. Some of the papers in this literature review were also discussed in previously published surveys. This table gives an overview, which papers have been reviewed in the respective surveys.

Category	Methods <sup>7</sup>
Survey	Shiravi et al. [216] [99] [180] [237] [243] [69] [209] [68] [14] [142] [149] [173] [245] [277] [244] [2] [49] [77] [143] [161] [1] [145] [160] [271] [70] [102] [155] [198] [91] [148] [199] [246] [50] [273] [165] [182] [22] [81] [131] [163] [189] [239] [276] [159] [215]
	Biersack et al. [25] [241] [18] [245] [49] [271] [55] [148] [181] [246] [214]
	Franke and Brynielsson [93] [191] [193] [19] [130] [71]
	Guimaraes et al. [108] [99] [180] [237] [243] [69] [209] [68] [14] [142] [149] [173] [245] [277] [244] [2] [49] [77] [143] [1] [145] [160] [271] [70] [102] [155] [198] [45] [91] [148] [199] [246] [50] [273] [165] [182] [22] [81] [131] [163] [172] [189] [239] [166] [240] [46] [119] [276] [24] [159] [215] [28] [140] [53] [115] [125] [168] [285] [186] [112] [179] [286] [90]
	Wagner et al. [261] [278] [185] [51] [195] [252] [105] [176] [196] [9] [106] [206] [275] [290] [109] [188] [272] [134] [63] [104] [162] [213] [267] [111] [110] [212]

### Stages for Situational Awareness

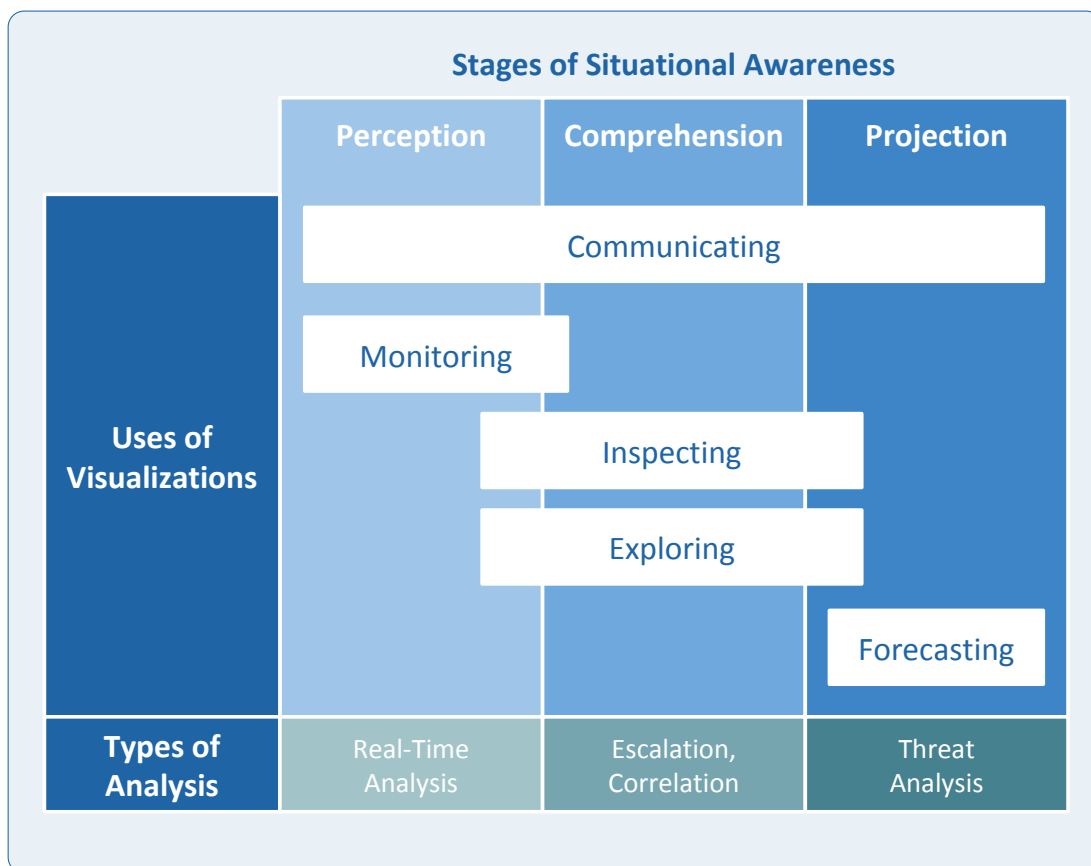
As discussed in the introduction of this chapter and summarized by D’Amico and Kocka [58], “*situational awareness is not a simple, atomic state: it is a process*” [58]. According to Endsley [67], situational awareness is based on three major stages: perception, comprehension, and projection [67].

- **Perception** – The stage of perception “*refers to the knowledge of the elements in the environment that one must know about, such as knowing what the Intrusion Detection System (IDS) alerts are*” [58]. Supporting this stage through information visualization could mean, that the analyst must be able to perceive the overall network activity, so that possible outliers or hosts with much network activity become visually apparent as for example in *ClockMap* [82].
- **Comprehension** – This stage “*refers to how people combine and integrate the elements they perceive, to derive meaning from them with respect to their goals*” [58], which can be described as “*knowing when you have perceived something important*” [58]. Visualization can for example help to enhance this stage to support the analyst in exploration of connected events resulted from an individual attacker. This often relates to highly interactive visualization systems, in which various views are available to analyze a given event from multiple perspectives. Extensive drill-down capabilities also help to foster comprehension through exposing the

underlying data to the analyst. These techniques basically provide evidence to support possible hypotheses of the analyst.

- **Projection** – The projection stage “*is the individual’s ability to project forward in time to anticipate future events. For example, mentally calculating that if the current sequence of suspicious events continues, and they are coming from the same source, then the next likely event will be of a specific type*” [58]. Visualization can help to visually analyze the threats to identify the modus operandi of similar attacks, to help the analyst to mentally project and predict likely future events.

These stages can be related to the various uses of visualization and the general analysis types. Both categories are discussed in the following sections. Figure 2.3 visually presents the general relationships between these categories. While visualization is for example needed for *communication* in all SA stages (e.g., to communicate results to other analysts or managers), *visual exploring* is more relevant in the *stage of comprehension*.



▲ **Figure 2.3** — Overview for stages of situational awareness. The general relations between *visualization usage*, *analysis type*, and *stage of situational awareness*. Table 2.3 gives an overview of trends for the various categories. *This modified and adapted figure is based on D’Amico and Kocka [58].*

## Uses of Visualization

D’Amico and Kocka [58] identified five general uses of visualizations for cyber security analysis, which are general enough to be applied to most of the literature and even to other domains.

- **Monitoring** – An analyst *“who is monitoring a system is watching an ongoing phenomenon in which data may be continually changing. It is part of the perception stage of situational awareness”* [58]. To always present the actual current situation, real-time aspects are important. Additionally, the visualization should be updated automatically as soon the underlying phenomena changes. This can be achieved using expressive dashboard designs [87] or even more sophisticated visualization systems as proposed in Chapter 5.
- **Inspecting** – It is obvious that the analyst wants to further inspect interesting situations perceived during monitoring. The *“analyst searches for specific details, requests clarification, and finds data to test hypotheses.”* [58]. *“Inspection is part of the perception stage of situational awareness, and may continue into the comprehension stage”* [58] when the analyst tries to further explain and judge the findings.
- **Exploring** – Besides of the specific inspection of interesting parts, the analyst is interested in exploration, which is *“characterized by undirected perusal, opportunistic discovery without a priori clues, novel data combinations, interactive experimentation with data views, finding data regions of interest for analysis, and hypothesis generation. Exploration relates to the perception phase of situational awareness when the analyst is striving to see patterns, and relates to the comprehension phase when he or she begins to explain the findings and assess the situation”* [58].
- **Forecasting** – *“The goal of forecasting can be to either find the likely future state presuming the current progression continues without intervention, or to determine a particular future state based on potential courses of action”* [58]. This is not necessarily done only by the integration of an automated analytical model, but also by manual *“pattern matching and trending”* [58]. Therefore, forecasting can also be done by the analyst using an implicit mental model. This is often *“achieved by matching the current situation against the past, and projecting the future based on past progressions.”* [58]. In the field of threat analysis and the investigation of the modus operandi during attack campaigns (attack attribution), visualization can also be used to forecast an emerging situation based on similar attacks with the same pattern in the past. This can be done by attributing the current situation to an already known attack campaign.
- **Communicating** – *“Visual data presentation is a useful means for communicating with other people, reporting to them, and educating them about one’s activities.”* [58]. Communication is relevant on all stages for situational awareness depending on the particular goal of communication. While decision-makers often rely on reports, it is important that the visual representation to communicate complex observations, are accurate and not misleading. Sometimes, it is indispensable that the visualizations still convey the context, so that a situation can be judged adequately.

### Types of Analysis

- **Real-Time Analysis** – In cyber security analysis there are different functions and roles. *“The ‘real time’ analyst may have as little as 90 seconds to make a decision regarding whether activity is suspicious or not. To support real time analysis, visualizations must automatically update with new data”* [58].
- **Escalation/Correlation** – In contrast with real-time analysis, many tasks, especially based on historic data, are related to escalation and correlation. For example, analysts who are dedicated to correlation, *“search through a day’s or week’s worth of data, often across many sites, looking for unusual trends to ‘pop out at them’”* [58]. Some tools especially focus on these scenarios. The *“popping-out that occurs is actually a cognitive event, when the analyst associates several pieces of information with each other and adds a hypothesis for why these events are all related. Data visualizations enables such ad hoc ‘visual discovery’ and recognition of patterns, trends, and anomalies”* [58].
- **Threat Analysis** – Some visualization systems explicitly help the analysts to analyze threats and attacks in a detailed way. They provide possibilities to identify common patterns, which is important to attribute an attack to a particular campaign or type of attack. The impact and detailed behavior of a malware sample is also highly interesting, because understanding such data helps to assess the threat.

### Use Case Classification

An important point of view to categorize and classify security-related visualization tools, is the intended use case. This is especially true for complex systems and also for visualization techniques that make use of heterogeneous data sources or can be applied to different data types. Shiravi et al. [216] introduce an established taxonomy of five general use case classes, which we extend with the categories of malware behavior and attack attribution, which are highly relevant for cyber security but were not in the focus of Shiravi’s work. Additionally, we categorize the resulting seven use cases into two general classes: (i) network activity and (ii) network threats. The use cases of the first category (internal/external, port activity, and host/server monitoring) focus on the analysis of network activity, which primarily includes network traffic, but also system log events, and alerts. These use cases are interesting for network planning, troubleshooting, identification of network issues, but also for intrusion detection to enhance the security. However, the second category (attack patterns, routing anomalies, malware behavior, attack attribution) focuses on specific network threats and the deep forensic analysis of attacks and the resulting anomalies.

---

<sup>8</sup> References which are emphasized using bold font actually refer to methods contributed by this dissertation, but have been previously published. However, for consistency reasons I decided to include them in this overview.

▼ **Table 2.3** — Overview of yearly trends for situational awareness. The table gives an overview about the number of methods with respect to stages for situational awareness, uses of visualization, types of analysis, and use cases. Only few visualization systems address the projection stage or focus on the communication of insights. Threat analysis and attack attribution use cases are also underrepresented in research.

Year	SA Stage			Usage				Analysis				Use Case				Year			
	Perception	Comprehension	Projection	Monitoring	Inspecting	Exploring	Forecasting	Communicating	Real-Time Analysis	Escalation/Correlation	Threat Analysis	Internal/External Monitoring	Port Activity Monitoring	Host/Server Monitoring	Attack Patterns	Routing Anomalies	Malware Behavior	Attack Attribution	
2002	6	4	0	6	4	4	0	0	6	4	0	0	0	2	1	3	0	0	2002
2003	3	2	0	3	2	1	0	0	2	1	0	0	0	1	1	1	0	0	2003
2004	7	8	1	7	6	6	1	0	3	8	1	2	1	1	2	2	1	0	2004
2005	14	9	0	14	9	5	0	1	11	8	1	2	1	1	8	2	0	0	2005
2006	7	6	1	7	6	5	1	0	6	6	0	0	0	0	4	4	0	0	2006
2007	4	2	0	3	3	2	0	0	2	2	1	0	0	0	4	0	0	0	2007
2008	10	11	1	8	11	9	1	0	7	11	2	2	2	5	1	1	2	0	2008
2009	6	7	1	3	7	7	1	1	3	7	0	1	0	0	5	0	2	0	2009
2010	6	5	2	4	5	6	2	0	3	5	2	0	0	2	4	0	0	1	2010
2011	8	7	5	5	7	8	5	0	3	8	4	1	0	2	2	0	3	2	2011
2012	18	16	5	13	18	15	5	2	12	18	6	0	1	8	3	3	5	1	2012
2013	16	12	5	7	17	12	5	0	6	17	5	1	1	5	4	1	5	0	2013
2014	16	11	8	5	17	11	8	0	5	17	10	0	0	6	3	0	7	1	2014
2015	7	7	0	6	7	6	0	0	5	7	1	0	0	5	2	0	0	0	2015

### Use Cases Related to Network Activity

We identified the following three use cases, which are more related to general network activity, because the intentions are not only related to actual threats and attacks, but also focus on managing and maintaining an overview about network utilization.

- **Internal/External Monitoring** – Computer networks provide the infrastructure, so that hosts and servers can communicate with each other. A traditional view for computer networks of organizations and companies is to focus on the internal versus external networks (e.g., Internet). A visualization system focusing on internal/external monitoring “*incorporates a display of internal hosts, but in relation to communicating external IPs*” [216]. Ball et al. [14] presents a good example for this category, called VISUAL [14], where the internal network is mapped to a matrix-based grid in which each individual cell represents a computer host in the network. Rectangles arranged outside the matrix represent external hosts, while lines between the cells and rectangles depict the network connections.

▼ **Table 2.4 — State-of-the-art overview based on primary use case.** The adapted and extended use case classification based on Shiravi et al. [216] helps to group the approaches into various distinctive general use cases. Each approach is assigned to a single use case category, which represents the primary use case respectively.

Category	Methods <sup>8</sup>
<b>Network Activity</b>	
Internal/External Monitoring	[14] [277] [70] [102] [27] [193] [240] [28] [78]
Port Activity Monitoring	[173] [1] [131] [239] [168] [226]
Host/Server Monitoring	[237] [69] [68] [149] [77] [163] [172] [189] [191] [94] [21] [24] [19] [140] [71] [82] [85] [115] [157] [169] [211] [221] [98] [112] [113] [121] [156] [79] [122] [224] [263] [144] [151] [281] [42] [90] [264] [12]
<b>Network Threats</b>	
Attack Patterns	[99] [180] [209] [142] [244] [2] [143] [161] [145] [160] [155] [198] [45] [91] [199] [50] [273] [165] [182] [22] [164] [81] [166] [101] [46] [119] [276] [47] [159] [215] [260] [53] [130] [125] [285] [30] [7] [289] [179] [286] [41] [92] [287] [183] [37] [133]
Routing Anomalies	[242] [243] [241] [18] [245] [244] [49] [271] [55] [148] [181] [246] [214] [84] [186] [25] [187]
Malware Behavior	[278] [185] [51] [195] [252] [105] [176] [196] [9] [106] [206] [275] [290] [109] [188] [272] [134] [63] [104] [162] [213] [267] [111] [110] [212]
Attack Attribution	[279] [73] [203] [253] [88]

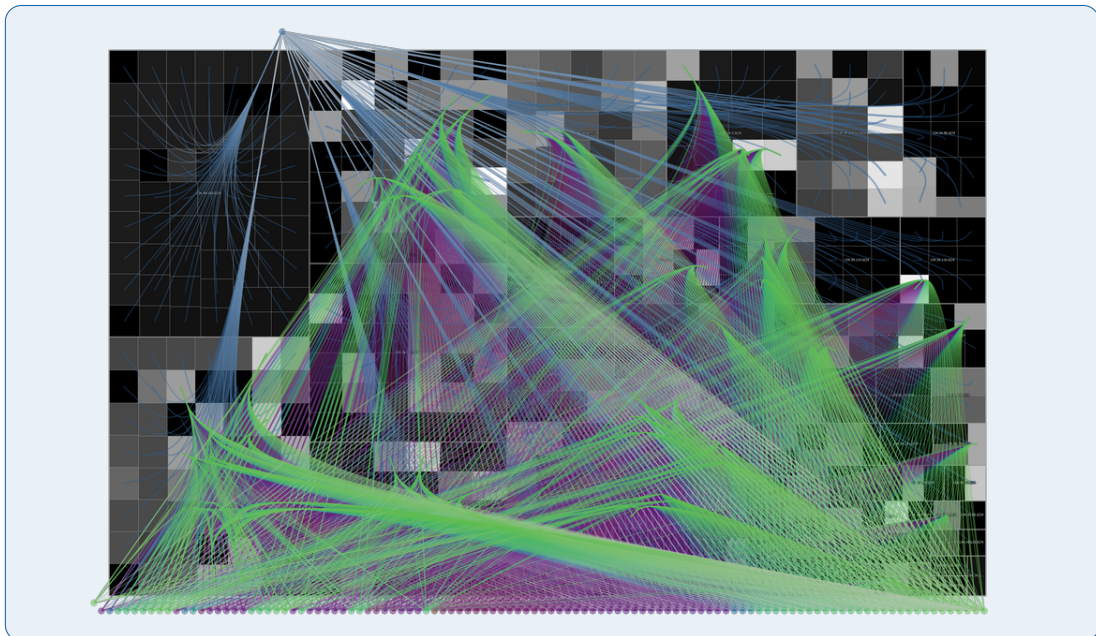
- **Port Activity Monitoring** – While network activity between hosts reveals interesting patterns and network utilization, the analysis of activity on particular ports provide a different perspective. Many services within the network operate on various standard TCP ports. High port activity on TCP/80 is most likely related to unencrypted HTTP web traffic. However, connections are not restricted to the well-known port numbers, but can occur on any arbitrary port as long the firewall permits such traffic. Observed port activity can help the system administrator to get an overview about services within the network, but can also reveal hints for compromised hosts, because some *“malicious programs like viruses, Trojans, and worms manifest themselves through unusual and irregular port activity”* [216]. However in our definition, we extend this category to include not only network activity time-series for various TCP ports, but also time-series data for other descriptors in a more generic way. For example, similarly to the number of packets on port TCP/80 over time, we have sensors in our network to count the number of packets related to a specific protocol (e.g., TCP, UDP, FTP, IRC). If we make use of deep packet inspection, the sensor, might even provide counters for in-depth packet information (e.g., HTTP request codes, browser versions, user agent strings, actual traffic type) or other specific keywords for which we want to provide a summarized activity overview.

- **Host/Server Monitoring** – This use case focuses on individual network hosts and servers within the computer networks. *“In this class of visualization, the main display is devoted to the representation of hosts and servers. The intent is to display the current state of a network by visualizing the number of users, system load, status, and unusual or unexpected host or server activities”* [216]. The display could focus on in-depth analysis of individual host and servers or provide a detailed overview for many network hosts.

### Use Cases Related to Network Threats

The following four use cases are stronger related to the inspection of actual threats, attacks, and anomalies. This also includes *routing behavior*, because the primary intent is to identify anomalies and BGP prefix hijacks which are severe network threats.

- **Attack Patterns** – *“Visualizations of this class aid an administrator in not only the detection of attacks but also the display of multistep attacks. Different types of attacks show different behaviors and accordingly different visual patterns appear”* [216]. In previous work as depicted in Figure 2.4, we proposed a visualization system called *NFlowVis* [81] to assess the relevance of alerts, to reveal attacks based on visual patterns. Analyzing the impact of intrusion detection alerts with such tools is important, to visually distinguish between false positives and critical alerts. Quick triage of intrusion detection alerts is crucial, because a *“major drawback of IDSs, regardless of their detection mechanism, is the overwhelming number of alerts they generate on a daily basis.”* [216].



▲ **Figure 2.4** — Visualization of attack patterns with *NFlowVis* [81]. The treemap represents the local computer network with hosts as rectangles. External attackers are shown as colored circles on the outside. The splines represent the connections between attackers and computers within the network. This reveals a network scan (from top) and a distributed attack (bottom) originating from hundreds of hosts.

- **Routing Anomalies** – Routing is a fundamental concept in the Internet. Correct path announcements are important to reach the correct destination servers. Despite of the importance and the severe consequences of routing issues, the responsible border gateway protocol (BGP) is quite vulnerable. Announcing malicious routing paths can be used to hijack IP blocks. As a result the attacker can conduct malicious activities from legitimate IP addresses. This highlights the need for visualization techniques to support this use case. *“Understanding the evolution of (...) routing patterns over time is the main goal of this visualization class”* [216].
- **Malware Behavior** – *“Malicious code (or malware) is defined as software that fulfills the deliberately harmful intent of an attacker. Malware analysis is the process of determining the behavior and purpose of a given malware sample”* [175]. This comprises static and dynamic malware analysis. In static analysis, the suspicious file is processed and disassembled to reveal common patterns, so that it can be distinguished from known, or identified as new malware family. In dynamic malware analysis, the malware sample is actually executed within a sandbox environment. Tools observe and log the behavior of all running processes. These behavior logs are then analyzed and compared to known characteristics. Both analysis approaches can benefit from visualizations. Visual analytics can also help to enhance situational awareness especially with respect to the projection stage, because knowing the capabilities of a given malware sample involved in a successful compromising attempt, helps to forecast and assess the consequences.
- **Attack Attribution** – This category involves the use case of attack attribution, which is *“primarily concerned with larger scale attacks (...) determining their root causes and (...) deriving their modus operandi”* [57]. Analysts try to relate attacks or malware samples to a larger group or attack campaign. Therefore, we define the use case of attack attribution with respect to visualization as, providing visual representations and visual analytics applications to explore and understand inter-related datasets and clusters describing large-scale attack campaigns. The overall goal is to relate new threats to a known group of attackers or campaigns, and to understand the modus operandi and trends within the threat landscape.

#### Attack Attribution or IP Traceback?

*“There is no real consensus on the definition of ‘attack attribution’ in the cyber domain”* [57]. David A. Wheeler and Gregory N. Larsen [61] define it as *“determining the identity or location of an attacker or an attacker’s intermediary”* [61], which is highly related to IP traceback, which can be defined as any *“technique that begins with the defending computer and recursively steps backwards in the attack path toward the attacker”* [61]. The ultimate goal of IP traceback is, therefore, to reveal the actual originating IP address, and eventually identify the real physical location and real-world identity of the attacker.

However, this dissertation follows the quite different definition by Dacier et al. [57] in which attack attribution is *“primarily concerned with larger scale attacks (...) determining their root causes and (...) deriving their modus operandi”* [57]. Thonnard et al. [249] also state while *“tracing back to an ordinary, isolated hacker is an important issue, we are primarily concerned by larger scale attacks that could be mounted by criminal or underground organizations”* [249].



## Data Sources

As also discussed by Shiravi et al. [216], there is a variety of different potential data sources for security visualizations, which can be categorized into more general event types. We use a similar but extended categorization to classify the most important data sources in the following classes: Network Traces, Security Events, User/Asset Context, Network Events, Host Events, Application Logs, and Malicious Data. A classification of which approach is applicable to which data source can be seen in Table 2.5.

- **Network Traces**

- *Packet Traces* – Packet traces are the actual full packets transferred over the network. A typical IP packet specifies various header information (e.g., version, length, protocol, checksum, source and destination IP address) and the payload, which is arbitrary data sent to the application.
- *Network Flows* – In large-scale networks, flow data is collected, because it is often not feasible to analyze the full packet traces. Routers, therefore, have the possibility to export meta data on a flow-based level. Many packet traces belonging to the same connection are aggregated. Flow packets (e.g., NetFlow) do not include any payload information, which is beneficial with respect to privacy. Most network traffic visualization tools actually rely on network flow records as seen in Table 2.5.

- **Security Events**

- *IDS/IPS Alerts* – Network intrusion detection systems (IDS) or intrusion prevention systems (IPS) like for example *SNORT*<sup>9</sup>, primarily analyze network traffic, while host-based intrusion detection systems (HIDS) such as *OSSEC*<sup>10</sup> also include specific host monitoring features and analyze log files, network traffic, and file system changes. They try to identify unusual and suspicious events primarily using rule- or signature-based approaches and generate alerts to notify the system administrators. An IPS provides active response capabilities to immediately block particular IP addresses or launch other countermeasures. Visualization helps to investigate, correlate, and explore such alerts to distinguish between false positives and critical events.
- *Firewall Logs* – Firewalls protect computer networks against illegitimate network traffic. Traditional stateful packet inspection (SPI) firewalls track the connection states of network connections, while deep packet inspection (DPI) goes various steps further and analyzes the actual packet contents to permit and block traffic based on application layer information. Firewalls provide capabilities to log information about successful or denied connections, which yield an important data source to be visually analyzed.

- **User/Asset Context**

- *Vulnerability Scans* – *Nessus*<sup>11</sup> is a popular scanner to identify potential vulnerabilities, misconfiguration, or weak passwords of remote systems within

---

<sup>9</sup> [www.snort.org](http://www.snort.org)

<sup>10</sup> [www.ossec.net](http://www.ossec.net)

<sup>11</sup> [www.tenable.com](http://www.tenable.com)

the network. These lengthy scan reports alone do not help, until they are analyzed by the system administrator. Eventually, the respective operators are responsible for closing the potential vulnerabilities on the various systems, before attackers start to actively exploit them.

- *Meta Data* – Some visualization systems also take a diversity of meta data into consideration. Meta data involves general information about the network structure, importance or usage of various hosts, or other network policies within the network. Additionally, meta data can be retrieved about potential attackers or involved domain names, by requesting registrar information, or geographical relation of a specific IP subnet. Common vulnerability and exposure (CVE) databases also provide important meta data information about known vulnerabilities and possible exploits.

- **Network Events**

- *System Metrics / Status Reports* – Popular tools like *Nagios*<sup>12</sup>, *Big Brother (BB)*<sup>13</sup>, and *Munin*<sup>14</sup> collect and monitor primarily system metrics from services and check their reachability. Such datasets provide rich information (e.g., CPU utilization, memory usage, latency, disk performance) about all services within large-scale computer networks. Such data reveal symptoms, which can help to identify hardware failures, configuration issues, or even security incidents. Abnormal and uncommon high CPU utilization on a particular web server could give hints about an ongoing distributed denial-of-service (DDoS) attack or could be a result of malware sending out huge amounts of spam e-mails.
- *DNS Logs* – Malicious software, which might be part of a larger botnet, needs to communicate with external command-and-control (C&C) servers. While IP addresses often change, some malware samples use sophisticated algorithms to generate special domain names. In a periodic manner these bots connect to the generated domain names and use them as potential rendezvous points. Such domain generation algorithms (DGA) were heavily used by “Conficker” and many other malware families. An attacker could just register one of these domain names in the future to communicate with the bots and control their behavior, because he knows all details of the algorithm and is aware of all possible domain names a bot will connect to. Such requests might be visible in the logfiles of the company’s domain name service, which is responsible for resolving requested domain names to valid IP addresses. Exploration of such DNS logs, therefore, can reveal suspicious or compromised machines in the network.
- *BGP Messages* – The border gateway protocol (BGP) is responsible for routing in the Internet. BGP messages convey the information how autonomous systems (AS) can reach specific IP prefixes. A router in the Internet can announce new routes, or withdraw previously announced routes. This update message (basically an IP prefix, along with a list of IDs reflecting the AS path) is sent to the neighboring autonomous systems. The receiving router

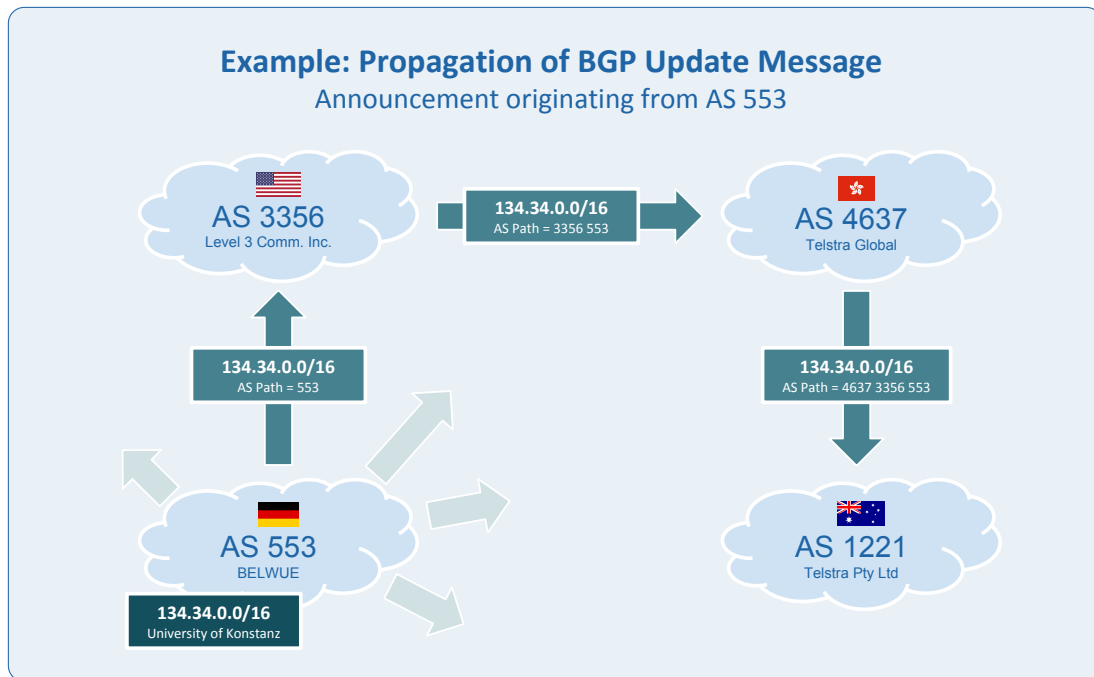
---

<sup>12</sup> [www.nagios.org](http://www.nagios.org)

<sup>13</sup> [www.bb4.org](http://www.bb4.org)

<sup>14</sup> [munin-monitoring.org](http://munin-monitoring.org)

updates the attached AS path and adds the ID of the sending AS to the path and again propagates the message to other routers as seen in Figure 2.5. Based on this information the routers update their routing table accordingly. The routers are then able to route packets to a specific IP prefix to the best neighboring router (with the shortest path) able to reach the final AS for the respective IP prefix.



▲ **Figure 2.5** — Propagation flow for BGP update messages. An example of a BGP announcement originating from AS553 to neighboring AS routers and incremental propagation to ASes around the world.

- **Host Events**

- *Server Logs* – Most hosts in the network provide logging capabilities. In large computer networks these log messages are normally forwarded to a central syslog server. This helps to centrally manage and correlate the events, but is also essential when a host crashes and can't be accessed any more. With a central logging infrastructure in place, chances are high that the stored syslog messages still reveal interesting hints to investigate the issue. The same is true in cases a host gets compromised by an attacker. Criminals often try to hide their traces and might remove various log files on the local machine. However, removing and hiding logs and traces on the central logging server is much harder, because they need to gain access first. Syslog messages normally contain arbitrary textual contents with an attached timestamp.
- *File System Changes* – Successfully compromised machines often have modified system files. An attacker who injects code on the server, or embeds malicious code to a website, leaves traces, which include modifications on the

file system. *Tripwire*<sup>15</sup> or *OSSEC*<sup>16</sup> monitor the whole file system or crucial folders for any file changes. Change reports can be analyzed to investigate suspicious unexpected file changes.

- *Audit Trails* – Audit trails or audit logs are highly related with the previous data source, but go beyond the simple change detection of files. Most operating systems and kernels can be configured to provide a detailed log about all system calls and operations done by the currently running processes. This involves file changes and the various file revisions, but also the initiation of network connections, execution of programs, or any other events a process is involved in. However, to identify actual suspicious or even malicious operations in these vast log files is quite challenging and often only feasible in the context of forensic analysis.

- **Application Logs**

- *Webserver Logs* – Within a computer networks there are various applications which provide extensive log files. A critical source for log files are webserver logs, because these server provide public data and information to external clients. Such webserver or proxy logs provide detailed information about which files have been requested. While most access requests are legitimate and might be initiated by actual users, there are various other clients. Many search engines heavily crawl all public websites, which results in a high amount of requests within the webserver logs. However, most of these requests can be easily identified using request parameters (e.g., user agent, IP range). However, there are also various other scanners trying to find vulnerabilities in the provided web services, or actively exploit known flaws. Monitoring such events and evaluating the successful attempts is crucial for situational awareness in cyber security.
- *Database Logs* – Databases provide various attack vectors for criminals. Databases contain crucial and often sensitive data, which is highly interesting for criminals involved in data breaches. Monitoring the access logs and queries initiated to databases helps to identify suspicious and unusual requests, or queries violating appropriate access rights or policies.

- **Malicious Data**

- *Honeypot Logs* – Honeypots are (virtual) services provided by special software applications. They are just used to attract intruders, but do not provide any real services. They are often set up in a way, that it is easy for attackers to gain access to these machines using weak passwords or well-known exploits. However, external attackers are not aware that they are just connected to a honeypot. The honeypots log all interactions done by the intruder. System administrators and security analysts, can then use the interaction logs, to analyze the attacker's behavior or understand the modus operandi. Furthermore, the intruder's IP address can immediately added to a black list preventing access to all real services in the network.

---

<sup>15</sup> [sourceforge.net/projects/tripwire/](http://sourceforge.net/projects/tripwire/)

<sup>16</sup> [www.ossec.net](http://www.ossec.net)

- *Spam / Phishing Mails* – Spam traps operate in a similar way than honeypots. These are just e-mail addresses used to capture all kinds of spam or phishing mails. These e-mail addresses are only set up to attract spam, and not for any legitimate communication. With the help of such spamtraps, it is possible to collect huge amount of malicious e-mails, or malware by extracting the attachments. This data can be used to gather new and previously unknown malware samples, or to identify IP addresses of compromised servers sending out spam. This information can be used to improve spam filters or help to understand the modus operandi of large-scale spam campaigns.
- *Malware Files* – The actual malware files collected via honeypots, spamtraps, or other means, can also be analyzed. Therefore, some visualization systems can read the binary files and relate them to other known malware samples.
- *Behavior Logs* – Malware can also be executed within sandbox environments to capture the behavior of a running malware sample. These behavior logs can contain system calls, network traffic, memory dumps, or any other interaction initiated by the malware. This content-rich information helps to extensively analyze the specific characteristics of malware families and judge their impact, which is important to assess the overall severity to enhance situational awareness with respect to the projection stage.

## Visualization Types

For the categorization of the different visualization techniques we used the *Information Visualization and Data Mining* taxonomy by Keim [136]. More precisely, we focused on the part discussing visualization techniques. Based on this taxonomy it is possible to divide the used techniques into five generalized categories:

- **Standard 2D/3D Displays** – Includes visualization techniques like *x-y* (*x-y-z*) plots (e.g., scatter plots), *bar charts*, and *line graphs* [136].
- **Geometrically-transformed Displays** – This category aims to visualize interesting transformations of multi-dimensional datasets (e.g., *scatter plot matrices* [10], *node-link diagrams*, *parallel coordinates* [136], *stardimates* [152]).
- **Iconic Displays** – The attributes of multi-dimensional data items are mapped onto the features of an icon or glyph. These compact representations are then mapped to the display (e.g., *chernoff faces* [43]), *needle icons*, *star icons*, *stick figure icons* [192], *color icons*, and *tile bars*).
- **Dense Pixel Display** – Each data point is mapped to a colored pixel and they are grouped into adjacent areas that represent individual data dimensions (e.g., *matrix visualizations*).
- **Stacked Display** – Representations for hierarchical data (e.g., *treemaps* [217]) and hierarchical layouts for multi-dimensional data (e.g., *dimensional stacking* [154]).

▼ **Table 2.5** — State-of-the-art overview based on primary data sources. This overview represents the primarily used data sources in the reviewed methods.

Category	Methods	
Data Source	Packet Traces	[99] [14] [244] [77] [1] [145] [70] [102] [155] [50] [273] [165] [182] [27] [131] [163] [189] [119] [260] [28] [53] [82] [125] [290] [226] [179] [92] [281] [37] [133]
	Network Flows	[149] [173] [277] [271] [198] [45] [164] [81] [131] [191] [193] [239] [166] [240] [101] [46] [21] [24] [260] [140] [82] [30] [78] [112] [226] [289] [41] [79] [92] [287] [90] [264] [183] [12] [133]
	IDS/IPS Alerts	[180] [209] [142] [2] [143] [161] [160] [155] [91] [22] [81] [163] [166] [276] [215] [19] [28] [125] [285] [221] [7] [98] [112] [121] [289] [286] [41] [79] [122] [287] [90] [12]
	Firewall Logs	[47] [130] [168] [285] [221] [98] [289] [286] [122] [90]
	Vulnerability Scans	[47] [130] [115] [144]
	Meta Data	[271] [91] [276] [47] [159] [279] [140] [203] [130] [168] [78] [41] [88] [92] [263] [287] [144] [90] [12]
	System Metrics / Status Reports	[69] [68] [172] [71] [82] [211] [113] [226] [289] [41] [263] [287] [144] [42] [90] [12]
	DNS Logs	[199]
	BGP Messages	[242] [243] [241] [18] [245] [244] [49] [271] [55] [148] [181] [246] [214] [84] [186] [25] [187]
	Server Logs	[237] [69] [68] [94] [85] [169] [121] [79] [122] [224] [151]
	File System Changes	[157] [156]
	Audit Trails	[159] [263] [267] [281]
	Webserver Logs	[7] [121] [122] [151]
	Database Logs	[121]
	Honeypot Logs	[260] [279] [73] [203]
	Spam / Phishing Mails	[84] [253] [88] [263]
	Malware Files	[278] [185] [51] [195] [252] [73] [105] [176] [196] [9] [275] [290] [109] [188] [272] [134] [63] [88] [162] [213] [111] [110] [212]
	Behavior Logs	[77] [195] [252] [159] [105] [196] [9] [106] [206] [275] [88] [104] [267]

▼ **Table 2.6** — Yearly trends for visualization types and techniques. The table gives an overview about the most widely used visualization types based on a general taxonomy by Keim [136] and various common visualization techniques.

Year	Visualization Type						Visualization Technique																		Year		
	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display	Glyph	Node Link	3D Node Link	Scatter Plot	3D Scatter Plot	Color Map	Treemap	Parallel Coordinates	Histogram	Timeline	Tables	Matrix	Geographic Map	Small Multiples	Word Cloud	Pixel Visualization	Radial Visualization	Chord Diagram	Sunburst Chart	Other Standard Charts	
2002	3	2	2	2	3	1	2	2	2	0	1	2	1	0	1	0	0	2	0	0	0	3	0	0	0	0	2002
2003	1	1	2	1	0	0	2	2	1	0	0	0	0	0	1	2	0	0	0	1	0	0	1	0	0	0	2003
2004	5	1	3	2	5	1	2	3	0	5	1	2	1	1	3	1	1	4	0	1	0	5	0	0	0	0	2004
2005	8	2	7	0	5	0	1	6	0	4	1	1	0	2	3	1	3	4	0	0	0	5	3	0	0	0	2005
2006	7	2	6	1	2	1	4	4	1	2	1	3	0	1	1	5	2	1	0	0	0	2	2	0	0	0	2006
2007	1	1	1	0	0	2	0	0	0	0	1	1	2	1	1	0	0	0	0	0	0	0	1	0	0	0	2007
2008	9	2	6	3	1	3	3	3	0	4	2	2	2	1	3	6	3	2	1	2	1	1	2	0	0	2	2008
2009	5	2	6	0	0	2	1	3	1	2	1	2	2	3	2	2	3	0	1	1	0	0	1	0	0	2	2009
2010	7	0	5	0	3	1	1	2	0	2	0	0	1	1	0	3	4	0	1	2	0	3	2	0	0	0	2010
2011	9	3	5	3	3	0	4	5	2	3	0	1	0	1	2	4	1	2	1	0	0	3	0	0	1	2	2011
2012	11	3	11	8	7	5	11	7	2	2	2	6	3	2	3	5	8	4	4	3	0	5	5	0	1	0	2012
2013	13	1	10	3	7	2	2	6	1	3	0	4	2	3	6	9	4	7	2	3	0	7	6	0	0	3	2013
2014	16	0	10	3	8	3	6	7	0	1	0	8	2	3	7	6	6	4	2	5	2	9	3	2	0	0	2014
2015	7	1	6	1	4	2	1	3	0	2	1	1	1	1	4	5	2	3	1	1	0	3	4	0	0	0	2015

## Visualization Techniques

Besides the aforementioned general visualization types, we also include a detailed analysis about concrete visualization techniques used in the various approaches. The overall yearly trends about these techniques are summarized in Table 2.6. The most popular visualization techniques were: node link diagrams, timelines, pixel visualizations, and glyphs.

## Interaction Support

For the categorization of the systems' interactive capabilities various interaction techniques such as zooming, filtering, panning, details on demand, or brushing/linking are available. Additionally, it is often possible to switch dynamically between different visual data representations. As seen in our study concerning malware visualizations [261], many of the papers did not specifically describe which of the aforementioned features they actually support. Most of the time, the tools were only dubbed as interactive in general without offering a more detailed explanation. Therefore, we decided to limit the categorization to whether the system supports any kind of interaction without going into detail.

▼ **Table 2.7 — Yearly trends for used evaluation techniques.** An overview about the most widely used evaluation techniques in the reviewed methods and applications. Obviously, case studies and usage scenarios are the most widely used technique, which we categorize as *insight-based strategies*.

Year	Evaluation Technique															Year		
	Field Study	Field Experiment	Longitudinal Study	Interview	Laboratory Experiment	Experimental Simulation	Judgment Study	Sample Survey	Formal Theory	Computer Simulation	Critique	Case Study	Usage Scenario	Competition Participation	Ground Truth Validation	Automated Image Analysis	Performance Testing	
2002	0	0	0	0	1	0	0	0	0	0	0	2	3	0	0	0	0	2002
2003	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	2003
2004	0	0	0	0	1	0	0	0	0	0	1	3	4	0	2	0	0	2004
2005	0	1	0	1	1	0	0	0	0	0	0	7	7	0	0	0	1	2005
2006	0	0	1	1	0	0	0	0	0	0	0	5	3	0	1	0	1	2006
2007	0	0	0	1	0	0	0	0	0	0	0	3	1	0	1	1	0	2007
2008	0	0	1	2	0	1	0	1	0	1	0	4	10	0	0	0	1	2008
2009	0	0	0	0	1	2	0	0	0	0	0	4	4	0	1	0	1	2009
2010	0	0	0	1	0	0	0	0	0	1	0	2	6	0	0	0	0	2010
2011	0	0	0	0	0	1	0	0	0	0	0	5	6	0	3	0	0	2011
2012	0	0	0	1	0	0	0	0	0	0	0	11	11	1	4	0	3	2012
2013	0	0	0	0	1	2	0	0	1	1	1	7	9	2	7	0	0	2013
2014	0	2	0	5	0	5	0	0	0	2	0	10	6	3	9	0	0	2014
2015	0	0	0	0	0	2	0	0	0	0	0	3	4	2	2	0	0	2015

## Evaluation Approaches

As main categorization for the evaluation approaches we follow the general strategy taxonomy by McGrath [171], which was further discussed by Carpendale [38] in the context of information visualization. Additionally, we assign various techniques common in cyber security visualization identified by Staheli et al. [223] to the respective strategies. An alternative taxonomy, based on seven goal-based scenarios is proposed by Lam et al. [150]. They also introduce and discuss an overview of various evaluation methods and methodologies [150].

### • Field Strategies

According to McGrath [171], the key characteristic of field strategies “*is that the behavior system under study is ‘natural’, in the sense that it would occur whether*



or not the researcher were there and whether or not it were being observed as part of a study” [171]. This means that field strategies share a high amount of realism.

- **Field Study** – Carpendale [38] makes it clear, that a field study “*is typically conducted in the actual situation, and the observer tries as much as possible to be unobtrusive. (...) Examples of this type of research include (...) case studies in industry. In this type of study the realism is high but the results are not particularly precise and likely not particularly generalizable. These studies typically generate a focused but rich description of the situation being studied.*” [38].
- **Field Experiment** – On the other hand, a “*field experiment is usually also conducted in a realistic setting; however, an experimenter trades some degree of unobtrusiveness in order to obtain more precision in observations. For instance, the experimenter may ask the participants to perform a specific task while the experimenter is present. While realism is still high, it has been reduced slightly by experimental manipulation. However, the necessity of long observations may be shortened and results may be more readily interpretable and specific questions are more likely to be answered.*” [38].

To also capture the reasoning process, these and other evaluation types with domain experts can also be structured as **Pair Analytics** [13] session, which is “*a method for capturing reasoning processes in visual analytics*” [13], but is more obtrusive. In such sessions “*verbal data about thought processes in a naturalistic human-to-human interaction*” [13] is gathered. One subject matter and one visual analytics expert are actively working together on a specific task, while their interactions are being logged and verbal communication transcribed and analyzed.

- **Longitudinal Study** – When a field study is conducted for a longer period, we would classify them as longitudinal field study. *Multi-dimensional In-depth Long-term Case studies (MILCs)* [219] as proposed by Shneiderman and Plaisant [219] would also relate to such longitudinal studies.
- **Interview** – Many studies involve interviews, in which a user “*is asked a series of structured or semi-structured questions to elicit knowledge regarding a particular topic, domain, or workplace*” [223]. While this method is also relevant within other strategies, in most cases the interviewee is domain expert in the respective field. This category also includes less formally defined interviews, like informal feedback sessions with the users or domain experts. We decided to also include similar qualitative methods into this category, like think-aloud protocols, in which users are asked to give direct feedback of their thoughts and actions during actual usage of the application.

## • Experimental Strategies

Compared to the previous strategy, field research “*has to do with whether the situation exists prior to and independent of the investigator, versus having been concocted by the researcher*” [171]. In the following experimental strategies, the situation is, therefore, not the typical field environment, but an experiment orchestrated by the researcher.

- **Laboratory Experiment** – “*In a laboratory experiment the experimenters fully design the study. They establish what the setting will be, how the study*

*will be conducted, what tasks the participants will do, and thus plan the whole study procedure. Then the experimenter gets people to participate as fully as possible following the rules of the procedure within the set situation*” [38]. Usability Testing [223] or a traditional *user study* to test specific design decisions of an application would be part of this category.

- **Experimental Simulation** – In this method “*the researcher attempts to achieve much of the precision and control of the laboratory experiment but to gain some of the realism (or apparent realism) of field studies. This is done by concocting a situation or behavior setting or context, as in the laboratory experiment, but making it as much like some class of actual behavior setting as possible*” [171]. This would refer to the *Simulation* category identified by Staheli et al. [223]. User studies, which are defined in a realistic way, to mostly reflect the actual field usage, would fall into this category.

- **Respondent Strategies**

These strategies “*concentrate on the systematic gathering of responses of the participants to questions or stimuli formulated by the experimenter*” [171]. These studies are arranged in a way, that they focus on observing behavior under conditions where the behavior setting is made irrelevant to the response [171]. These strategies are often applied when the results should be highly generalizable with high precision, but less focus is given to realism.

- **Judgment Study** – “*In a judgment study the purpose is to gather a person’s response to a set of stimuli in a situation where the setting is made irrelevant. (...) Ideally, the environment would not affect the result. Perceptual studies often use this approach.*” [38]. This category is also related to a *Laboratory Experiment*, however the focus is more focused on a particular set of stimuli. Studies in this category sometimes make use of psychophysiological measurements (e.g., brain activity).
- **Sample Survey** – “*In a sample survey the experimenter is interested in discovering relationships between a set of variables in a given population*” [38]. This covers general questionnaires sent to a larger group of persons answering specific questions about a given topic or prototype.

- **Theoretical Strategies**

- **Formal Theory** – “*Formal theory is a strategy that does not involve the gathering of any empirical observations*” [171]. Carpendale [38] gives an example for a formal theory, in which “*the results of several studies can be considered as a whole to provide a higher-level or meta-understanding or the results can be considered in light of existing theories to extend, adjust or refute them*” [38]. The work by Alshaikh et al. [8] could be considered in this context as well, in which security applications are evaluated using the theories of Alexander [6], which can be summarized in fifteen properties of order [6]. Alshaikh et al. [8] then discuss in which extent the various security applications follow the suggested properties of order. Therefore, the authors use properties based on various formal theories to evaluate and compare the strengths and weaknesses of various approaches.

- **Computer Simulation** – This is another theoretical strategy, which is non-empirical. *“It is like the experimental simulation strategy (...) in that it is an attempt to model some particular kind of real-world system”* [171]. In the scope of cyber security visualizations, this could be an evaluation based on an automatically generated synthetic dataset to be used in a case study. For example, Fowler et al. [92] *“generated synthetic DDoS attacks of varying intensity against a monitored network over a period of 25 minutes, using several attack topologies”* [92]. Such data generation would relate to an evaluation based on a computer simulation, because the data is generated synthetically to reflect real-world scenarios.
- **Critique** – This methods builds around *“a meticulous group discussion centered on how well particular aspects or details of a visualization support the intended goal”* [129]. This method also relates to formal theory, because critique and comments by the participants of group discussions, are often based on previous perceptual studies.
- **Inspection** – *“Usability inspection is the generic name for a set of methods that are all based on having evaluators inspect the interface (...) aimed at finding usability problems”* [178]. According to Nielsen [178] this includes for example, heuristic evaluation, cognitive walkthroughs, feature inspection, consistency inspection, standard inspection, and others. While strictly speaking this is not a traditional theoretical strategy, it still has some overlap, because such techniques (e.g., heuristic evaluations) often follow known theoretical and *“established usability principles”* [178].
- **Insight Strategies** – We introduce this category, which is not part of the taxonomy by McGrath [171], to focus on evaluation techniques directly related to describing the identification of findings and insights in visual analytics applications, which is often done by various case studies or ground truth validations.
  - **Case Study** – There are various types of case studies. Isenberg et al. [127] distinguish between (i) case studies from domain experts, (ii) case studies from close collaborations, and (iii) case studies conducted by visualization researchers, who are also experts in the problem domain. Because of the lack of details provided by the authors, it is often hard to assign the described case studies to one of the stated types. Therefore, we decided to keep all three types in a single category.
  - **Usage Scenario** – This category is often also described as case study by the authors, however “case studies” which only report how the visualization approach *could be used* by hypothetical domain experts [127] are better described as use cases or usage scenarios.
  - **Competition Participation** – van Wijk [255] highlights the challenge of evaluation for visual analytics application, but also state that an *“interesting alternative approach to evaluation is competition: present a problem to the community and challenge researchers and developers to show that their solution is best”* [255]. Active participation in international competitions and challenges help to compare and evaluate many approaches.
  - **Ground Truth Validation** – The aforementioned challenges and competitions can also help to orchestrate ground truth data, to validate findings with

actual known findings. However, there are also other benchmark datasets available to validate the gathered results. This often can be done for malware classification use cases, in which known training data is available. However, it is much harder for general network traffic datasets, because for real datasets there is often no ground truth available at all.

- **Computational Strategies** – Sometimes it is also possible to follow fully automated computational strategies, in which no users are directly involved. The focus of such techniques is more the evaluation of scalability with respect to performance or processing time, but also the automated analysis of visualization results with respect to various defined optimization functions.
  - **Automated Image Analysis** – “*Computer-generated analysis of a digital image for visual characteristics*” [223] is the core of such analysis techniques. For example, Mansmann et al. [165] use an optimization function to evaluate the automatically generated data-driven layout adaption of treemaps with respect to visibility, average aspect ratio, and layout preservation [165].
  - **Performance Testing** – Testing the performance, memory usage, or other resources, is a common technique to validate the technical scalability of a system or a complex method. This also refers to the discussion of an algorithm’s complexity, which is often important to judge the applicability of a method to real-world scenarios. However, such techniques often do not help with respect to the actual usefulness of an approach to answer analyst questions. Therefore, performance testing is often used together with case studies to cover the evaluation of usefulness and scalability issues.

## 2.2 Observations and Research Gaps

In the following we summarize *Observations and Research Gaps* (OG)<sup>17</sup> based on our extensive literature review for cyber security visualizations, which lead to specific research objectives to be addressed in this thesis.

**OG1 No single holistic visual analytics system for cyber security SA** – Gaining situational awareness in cyber security is indeed challenging. Each reviewed system, visualization, and application, only contributes to single aspects in the situational assessment workflow. There seems to be no holistic solution to address all questions with respect to all use cases. While it would be desirable to have a single system to address all aspects, more interdisciplinary research has to be conducted to reach this goal.

**OG2 Supporting the SA projection stage is challenging** – Situational awareness relates to various stages. Visualization can be used during situational assessment to get a clear picture of the ongoing network activity and threats. This reaches from

---

<sup>17</sup>Please note, that parts of this dissertation have already been published at various venues to share results with the community as soon as possible. This is the reason, why the presented literature review does also contain some of our own applications and methods, which will be described in the following chapters of this dissertation. To provide an holistic view of available literature including publications which formed the basis of this dissertation, I decided to include them. However, the observations and identified research gaps are partly solved and addressed by our proposed methods.

basic perception of various interesting or suspicious findings, over comprehension, to actually judge the identified events, to a projection stage, in which the analyst gets a clear mental picture about the possible impact or consequences with respect to the future. While most analysis system directly support perception and comprehension, it is much harder to explicitly support situational assessment with respect to the projection stage. This is also reflected in the number of methods categorized to the various stages (Table 2.3), where only 29 are categorized to the projection stage. Most systems in this category, provide the analysis of specific attacks or malware samples. Being capable to classify an unknown malware sample to an already known class or family using visual analytics techniques directly supports the projection stage, because the analyst is then able to judge the likely impact and project it to possible consequences. However, on the other hand, systems with extensive visual exploration capabilities can also indirectly support the projection stage. The in-depth exploration and analysis of a specific attack pattern, can help the analyst to get a better understanding of the current threat.

- OG3 Not enough research for all primary use cases** – In the last decade most research was conducted for attack pattern visualization (45 methods). However, only 5 publications relate to port activity. 4 articles relate to the field of attack attribution to understand the more general threat landscape. 15 methods relate to routing anomalies and BGP prefix hijackings. Also various obvious reasonable combinations of heterogeneous datasets are seldom used. While many tools combine network flow data with IDS/IPS alerts (Table 2.5), there are no systems, which combine control-plane data (BGP messages) and observations related to data-plane data (e.g., spam and phishing mails).
- OG4 Limited support for context-aware inspection and exploration** – Most cyber security applications use visualizations for inspection and exploration. Smooth switching between these phases is often not intended. However, the underlying tasks are indeed often highly interconnected to gain good situational awareness. An analyst generally perceps interesting behavior during monitoring, which makes it important to inspect this behavior in more details. However, also exploring the context in which these anomalies happen is important to get a clear picture of the event. This context-dependent in situ inspection and exploration relates to the overall network, but also to the change over time during monitoring.
- OG5 Limited scalability of visualization techniques** – In general, there is lack of good scalability for many proposed visual techniques. While many systems are very helpful for small datasets, they are often not suited for real-world scenarios. Sometimes neither the visualization, nor the used implementation would be capable to address the current data load. Especially, many visual techniques do not provide scalable exploration of the vast number of computer hosts in today’s networks.
- OG6 Limited usage of novel scalable analytics methods** – While most academic methods and systems still use traditional databases or self-developed analysis modules – not proven to be reliable for large-scale processing – only very few academic systems actually employ and make use of emerging scalable analytics products from the big data analysis domain (e.g., *Hadoop*, *Cloudera Impala*, *Apache Spark*, *Apache Storm*) or hosted large-scale IaaS (Infrastructure as a Service) systems (e.g., *Google BigQuery*).

**OG7 Only few dynamic visualizations for real-time monitoring** – While many systems use visualization for monitoring and fall into the scope of real-time analysis, only few visual analytics systems (e.g., [27, 18, 203, 169]) actually address the challenge of dynamic visualizations to be updated in real-time. Many systems claim to be used for monitoring and real-time scenarios, however with heterogeneous data sources the underlying visualization techniques often provide no possibilities to be updated automatically and circumvent the resulting challenges. Such systems normally rely on the user, to refresh the current view or select a temporal period, which interrupts the visual analysis and might lead to a complete change of the visualization.

**OG8 Evaluating complex cyber security applications is challenging** – The proper evaluation of visual analytics application in general is extremely challenging. As stated by van Wijk [255], “*Insight, the major aim for visual analytics, is ill-defined and hard to measure*” [255]. This is true for general visual analytics applications, but even more for the complex domain of cyber security, in which it is hard to establish ground truth data, because of privacy issues, confidentiality, data volume, and continually evolving threats. According to Table 2.7, most reviewed systems only employ usage scenarios to evaluate their methods. This trend is also observed by Staheli et al. [223], who state that such an evaluation technique “*can serve a purpose, but it is important to note that a lack of connection back to real users or real data (or both) may question the validity and utility of the evaluated tool*” [223]. While field strategies provide direct contact with real experts to evaluate the methods, they are often not feasible in the security domain. Another approach, which is quite promising is the active participation in competitions, which “*have helped advance some fields quickly*” [255].

**OG9 No focus on communicating findings in VA application** – Evaluating the main uses of visualization (Table 2.3) reveals an interesting trend. Surprisingly, there seems to be no focus on communicating findings in cyber security visual analytics. Only very few systems, provide visualizations with an *explicit* focus to communicate hypothesis, findings, and results to others. *VIAssist* by Goodall and Sowul [101] is one of the few visual analytic systems integrating reporting functionalities. The generated reports help others to understand the ongoing situation easily through visual representations. The focus of these visualizations is not to identify new insights, but to share and communicate suspicious patterns or the current situation with others. However, most visualizations could be used for communication reasons (as screenshots) as well, but only few visual analytics systems actually focus on such a usage.

## 2.3 Research Objectives

While it is still a long way to close the gap and build a holistic situational awareness system for all aspects of cyber security (OG1) and eventually anticipate or even predict future threats (OG2), it is in the meanwhile important to further develop and improve methods for *all* primary use cases in cyber security (OG3). Addressing OG3 will eventually lead the way to the vision of an holistic visual analytics application. Based on the aforementioned observations, we want to address some of the discovered research

challenges and achieve the following specific *Research Objectives* (RO), which pave the road towards closing the identified gaps (OG4-8):

**RO1 Introduce novel visual techniques for context-aware exploration to support visual analytics for network activity** – Exploration of temporal series within its context is important for network activity analysis. However, most systems provide limited visual support (OG4) and are often not scalable enough (OG5). It is, therefore, important to investigate novel techniques which inherently provide context-aware exploration, for example, to correlate a given host with neighboring hosts in the same network. Such techniques, help to spot outliers based on visual patterns. Such approaches can be evaluated in various ways, using laboratory experiments, insight-based strategies, or active participation in contests to compete with others (OG8).

*This research objective will be addressed mostly in Chapter 3, in which we introduce a scalable visual analytics system for cyber security, called VACS, which is evaluated through active participation in a cyber security visualization challenge (OG8). Furthermore, we extend this work with correlation analysis of time-series within the context of similar and correlated series (OG4) using IAS-Explorer. Additionally, we introduce ClockMap, which is a novel and scalable hierarchical glyph-based technique to investigate time-series of network hosts within the context of different network subnets (OG4). This technique is evaluated and integrated into BANKSAFE, backed up by a scalable IaaS backend (OG6), to participate in an international visual analytics competition (OG8) to solve realistic cyber security tasks.*

**RO2 Combine multiple data sources to improve SA for BGP routing** – Attacks on the control plane (e.g., BGP prefix hijacking) are a major security threat in the Internet. While there is some research in the area of BGP analysis, none of the reviewed system include multiple data sources, to combine control- with data-plane analysis (OG3).

*This research objective will be addressed in Section 4.2, in which we introduce VisTracer, a novel visual analytics application combining IP traceroutes from ongoing spam and phishing campaigns to correlate BGP routes with malicious activity (OG3).*

**RO3 Integrate visual analytics techniques for attack attribution** – Based on our literature review, most research was conducted in attack pattern visualization for intrusion detection. While this is an important field of research, it is also indispensable to understand the modus operandi of attackers on an higher level. How do the criminals operate? To which attack campaign, does a current attack belong to? A good understanding helps to estimate or project the actual threat level on a broader scale. Such an understanding is important to achieve good situational awareness. However, the literature review reveals a huge research gap with respect to visualizations supporting such attack attribution use cases (OG3). Therefore, we see it as essential to also support the analyst in the visual analysis of the threat landscape, to help understanding and projecting (OG2) such attacks with respect to orchestrated campaigns on a larger scale.

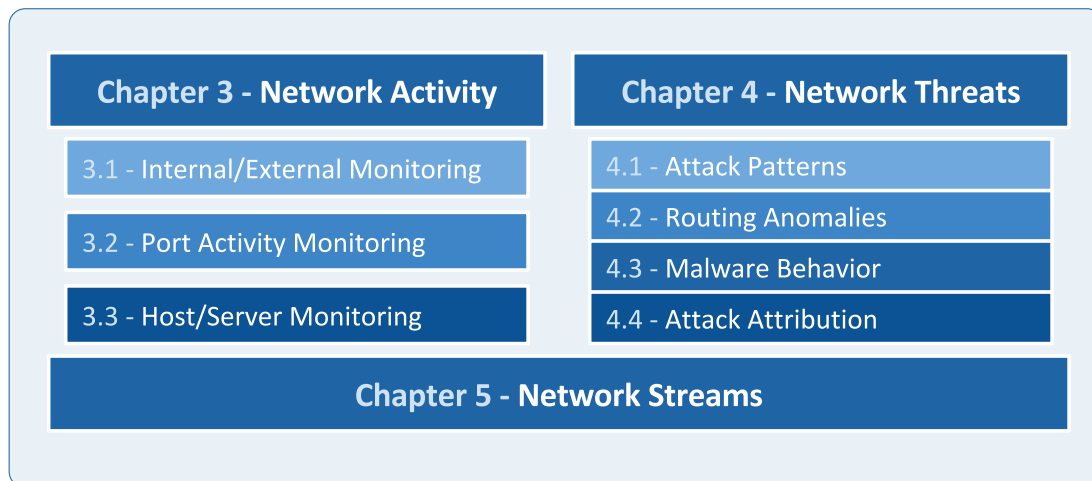
*This research objective will be addressed in Section 4.4 in which various visualizations are used to support visual exploration of clustering results generated by*

one of the leading algorithms for attack attribution (TRIAGE [249]). We include various visualization techniques into VACS, to eventually approach experts in a field experiment (OG8).

**RO4 Introduce a novel dynamic visualization concept for scalable real-time monitoring for heterogeneous data streams** – To specifically address research gap OG7, a visualization system for real-time monitoring is needed, which provides an interface updated in real-time, but still conveys the temporal context (OG4) for heterogeneous cyber security data streams. Because of the vast amount of data to be processed in real-world scenarios, it is important to make use of novel and scalable analytic methods (OG6). To address the challenge of evaluating the usage (OG8) of such systems for *general* situational awareness, active participation in competitions focusing on real-time event monitoring seems promising.

*This research objective will be addressed in Chapter 5 with its concepts shown in the NVisAware visualization method, and integrated to the scalable NStreamAware system. Furthermore, we apply the visual analytics approach to provide context-aware summarization, steered by the expert knowledge of the analyst to provide scalable methods for long time spans (OG5).*

In the following, based on the two main classes *network activity* and *network threats* as seen in Figure 2.6, we discuss in Chapter 3 and 4 the various use cases relevant for cyber security visualizations and propose solutions to fill the aforementioned research gaps respectively. To contribute explicitly on visual analytics for real-time use cases, we use Chapter 5 to focus on situational awareness aspects for network streams.



▲ **Figure 2.6** — Thesis structure based on cyber security use cases. Visual structure overview of the main chapters along the various network security use cases.



*“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards.”*

— Gene Spafford

C H A P T E R



3

## Visual Analytics for Network Activity

### Contents

---

<b>3.1</b>	<b>Visual Overview for Internal and External Monitoring . . .</b>	<b>40</b>
3.1.1	VACS – Visual Analytics Suite for Cyber Security . . . . .	42
3.1.2	Evaluation using VAST Challenge 2013 . . . . .	48
3.1.3	Conclusions and Limitations . . . . .	57
<b>3.2</b>	<b>Visual Correlation for Port Activity Monitoring . . . . .</b>	<b>58</b>
3.2.1	IAS-Explorer – Visual Analytics for Port Activity Correlation	61
3.2.2	Evaluation using Port Correlation Case Study . . . . .	65
3.2.3	Conclusions and Limitations . . . . .	68
<b>3.3</b>	<b>Visual Exploration for Host and Server Monitoring . . . . .</b>	<b>69</b>
3.3.1	ClockMap – Visualization Technique for Host Monitoring . . . . .	71
3.3.2	Evaluation of Alternative Glyph Designs . . . . .	77
3.3.3	Evaluation of ClockMap’s Design Principles . . . . .	80
3.3.4	Evaluation using VAST Challenge 2012 . . . . .	82
3.3.5	Evaluation using VAST Challenge 2013 . . . . .	90
<b>3.4</b>	<b>Conclusions . . . . .</b>	<b>100</b>

---

NETWORK activity use cases are often related to various types of network monitoring. However, these techniques are not only used for monitoring, but also for in-depth exploration with different view points. Therefore, we divide this category into three use cases: (i) internal/external monitoring, (ii) port activity monitoring, and (iii) host/server monitoring. All these areas together with interactive exploration are eventually needed to get a clear picture about ongoing network activity to contribute to the state of situational awareness. This is not necessarily needed only for threat, attack, or intrusion detection, but also for management issues and maintaining an overview of the usual network utilization. However, this also relates to cyber security, because

successfully compromised computer systems often reveal network activity patterns, which are different from non-compromised machines.

In this chapter, we introduce various visual analytics systems, but also novel visualization techniques. In Section 3.1, we propose a system, called *VACS* to analyze internal network activity using a compact visual overview of individual time-series in a small-multiple setting and interactive node-link diagrams to explore external communication patterns. To automate the identification of interesting parts of the time-series, we provide means to support the analyst with visual correlation capabilities combining automated and explorative techniques making use of vertically oriented line charts in Section 3.2. Eventually, we contribute and discuss in Section 3.3 a novel visualization technique, called *ClockMap* to analyze temporal network data within a given hierarchy in a scalable interactive way.

### 3.1 Visual Overview for Internal and External Monitoring

Network activity analysis can be addressed from the perspective of internal versus external networks. Traditionally, many organizations and companies use routers, firewalls, and network address translation (NAT), to separate the internal local-area network (LAN) from the outside world (e.g., the Internet). Consequently, the distinction between internal and external computers is also quite common with respect to monitoring of network activity, because of this topological layout.

However, in the last years, this strong distinction (especially with respect to trustworthiness of internal hosts) became problematic, because a large percentage of successful attacks can be classified as insider threats, or were actually conducted over an internal host, which got compromised. When the initial attack of this internal host was conducted using a carefully crafted e-mail to an individual employee, chances are high that the receiver opens the e-mail without getting suspicious about the attached malware file. This is even more likely, when attackers employ social engineering techniques to gain trust of the victims. The fact, that such attack vectors can hardly be recognized in internal/external-focused monitoring visualizations, might be a reason why there are only few recent systems actually focusing on such use cases as seen in Table 3.1. Another reason, why only few visualization systems focus on this case, is the sheer number of legitimate external hosts in communication with the internal network hosts. To visualize these actual endpoints without applying clustering is often not feasible for large-scale networks. While in IPv4 networks it is possible to represent  $2^{32}$  (approximately 4.3 billion) IP addresses, the emerging IPv6 protocol, which was developed to replace IPv4 in the future, an unbelievable amount of about  $2^{128}$  addresses can be addressed. Taking these numbers into account the individual visual representation of these external endpoints is hardly feasible and does not scale for monitoring use cases. However, for attack visualization as discussed in Chapter 4, the presentation of individual external hosts can still provide valuable insights, because in most cases only a limited number of hosts are involved in an attack.

#### Related Work

Ball et al. [14] presents a good example for this category, called *VISUAL*, where the internal network is mapped to a matrix-based grid in which each individual cell represents a computer host in the network. Rectangles arranged outside the matrix represent external hosts, while lines between the cells and rectangles depict the network

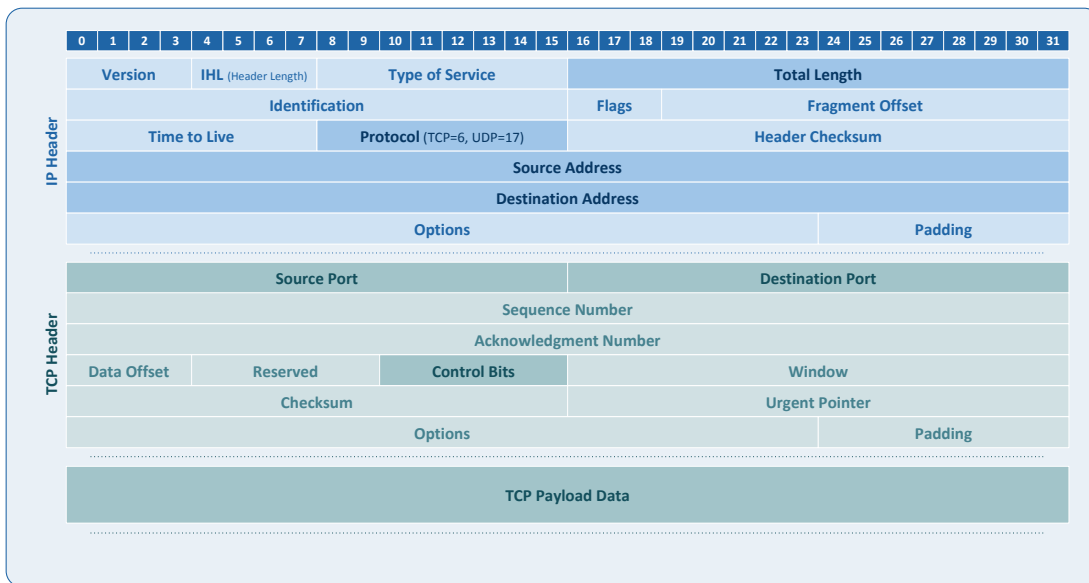
connections. With the help of such and similar representations the analyst can identify interesting network activities.

▼ **Table 3.1 — Related work for internal/external monitoring.** Overview of related work with respect to data source and visualization type.

Method	Use Case			Data Source													Visualization					Year						
	Internal/External Monitoring	Port Activity Monitoring	Host/Server Monitoring	Packet Traces	Network Flows	IDS/IPS Alerts	Firewall Logs	Vulnerability Scans	Meta Data	System Metrics / Status Reports	DNS Logs	BGP Messages	Server Logs	File System Changes	Audit Trails	Webserver Logs	Database Logs	Honeypot Logs	Spam / Phishing Mails	Malware Files	Behavior Logs	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display	
VISUAL [14]	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	2004
VisFlowConnect [277]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2004
Erbacher et al. [70]	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2005
TNV [102]	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	2005
NetGrok [27]	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	2008
NUANCE [193]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	2008
FloVis [240]	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	2009
TVi [28]	✓	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	✓	-	2011

From Table 3.1, we observe that most tools and methods use packet traces or network flow data as data sources. While the analysis of raw packet contents is often not possible because of privacy and performance reasons, the systems normally extract and only make use of the information available in the packet header. Figure 3.1 shows the header fields of a TCP/IPv4 packet. The length of the packet headers is a multiple of 32 bits as indicated at the top in Figure 3.1. The length of the IPv4 [59, page 11] packet header is  $5 * 32$  bits, plus the optional options and padding. The respective header format for IPv6 packets is defined in RFC 2460 [107, page 4], which we won't describe in detail, because IPv6 is seldom explicitly in focus of the surveyed systems. Most of the reviewed systems still focus on IPv4 related datasets. The encapsulated TCP [60, page 15] segment contains 10 mandatory header fields and the actual TCP payload data, which is transferred to the application. As highlighted in Figure 3.1 (emphasized with bold font style), most visualization systems only focus on payload/packet length, protocol, source/destination IP address, source/destination port, and control bits, while all other fields are less relevant for internal/external monitoring use cases. The analysis of the actual content (TCP payload data), is often done automatically based on signatures for intrusion detection, while visualization is used to explore the resulting alerts.

Alternatively, many monitoring approaches rely on network flow data (e.g., Cisco NetFlow [233], sFlow [54], IPFIX [234, 124]). According to RFC 7011 [124], in which the *IP Flow Information Export* (IPFIX) format is defined, the following flow definition is used: “A *Flow* is defined as a set of packets or frames passing an *Observation Point* in the network during a certain time interval. All packets belonging to a particular *Flow* have a set of common properties.” [124, page 8]. These properties primarily include IP and TCP header fields as seen in Figure 3.1, and various other characteristics [124].



▲ **Figure 3.1 — Overview of TCP/IPv4 packet headers.** The figure shows the header fields for an IPv4 packet and the encapsulated TCP segment.

The full list [123] of information elements as defined by IPFIX covers over 400 fields, which can be defined on such template-based IPFIX flow exporters. For the actual analysis most systems in our survey only make active use of a very limited number of information elements.

### 3.1.1 VACS – Visual Analytics Suite for Cyber Security

The sections coming next mostly build on the following publication [78]<sup>1</sup>:

F. Fischer and D. A. Keim. VACS: Visual Analytics Suite for Cyber Security - Visual Exploration of Cyber Security Datasets. In *VAST Challenge 2013 - Honorable Mention*, 2013 [78].

The perspective of an analyst monitoring computer systems is the internal network related to the external network. To stay focused to the highly relevant aspects, we address the challenge to propose an analysis workflow: (i) dashboard overview, (ii) temporal selection, (iii) selection of internal hosts, and (iv) exploration of related external connections using interactive node-link diagrams and treemaps. This allows a classical internal/external exploration of large computer networks, while related work is often not scalable enough and focuses only on very general patterns or very specific limited datasets. We implement this workflow in *VACS*, which is a novel visual analytics

<sup>1</sup> Within the VIS-SENSE project, I had the idea and implemented a web-based system to support visual exploration of temporal network data to enhance cyber security, to integrate various techniques developed in the project and make them applicable to other datasets to participate in the security-related VAST Challenges to compare with others. The challenge submission and the supplementary paper [78] were written by myself, while Daniel Keim gave advice and suggestions within the project.

suite to analyze and visually explore large-scale cyber security datasets. To achieve scalability for large datasets *VACS* makes use of an *ElasticSearch*<sup>2</sup> cluster with multiple nodes using commodity hardware. *VACS* is a web application using JavaScript, HTML5 and a variety of state-of-the-art toolkits and custom widgets and a mix of interactive client-side visualizations and visual representations generated on the server-side due to performance reasons. In the following, we describe the different elements and explain a basic use case how an analyst can use the system for visual exploration which can lead to a better situational awareness. Additionally, we successfully evaluate the system by active participation in the VAST Challenge 2013 Mini-Challenge 3 [269] to compete internationally with other researchers to solve the given tasks with respect to internal and external monitoring for situational awareness.

### Dashboard Overview

There is a lot of research in the design of information dashboards. Good guidelines are given by Few [75] in his book about information dashboard design. Many dashboards fail, because they do not convey the proper information and do not use appropriate visualization techniques. While we proposed some experimental concepts [87] towards user aware adaptiveness dashboards to investigate possibilities to automatically adapt a dashboard with respect to user awareness, we do not focus on such topics in this thesis. For *VACS*, we decided to include two types of widgets to monitor various important network metrics. Figure 3.2 shows an example of such a dashboard, in which only few metrics are shown. The dashboard uses so-called *bullet graphs* [74], to show the number of flows within the last hour, the number of flows related to HTTP traffic, and the unique count of destination ports. Stephen Few developed the bullet graph technique to “replace the meters and gauges that are often used on dashboards. Its linear and no-frills design provides a rich display of data in a small space, which is essential on a dashboard. Like most meters and gauges, bullet graphs feature a single quantitative measure (...) along with complementary measures to enrich the meaning of the featured measure” [74]. For example, in Figure 3.2 the “complementary measure” indicated as small white triangle could be used to represent the mean value of the respective measure. Additionally, the integration of an temporal histogram shows the general amount of network traffic over time as quick reference to identify peaks or overall trends.

### Temporal Selection using Interactive Line Charts

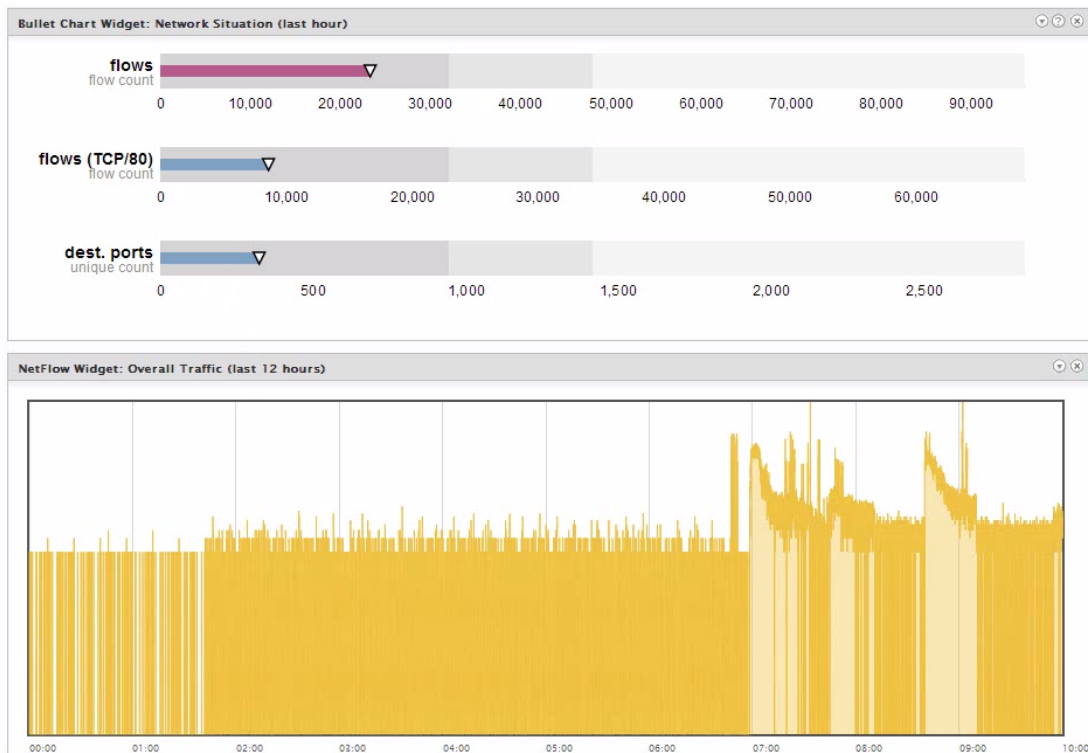
Line charts are a well-known visual representation for time-series exploration. The analyst can use a dialog to query the different datasets to extract time-series (e.g., network traffic over time, alerts above a threshold, traffic on specific ports, average memory consumption). This representation as seen in Figure 3.3 helps to correlate different time-series. However, the chart is also primarily used to guide the drill-down process to parameterize other visualization with the selected time interval to follow the proposed workflow.

### Overview of Internal Hosts using Striped Thumbnail Glyphs

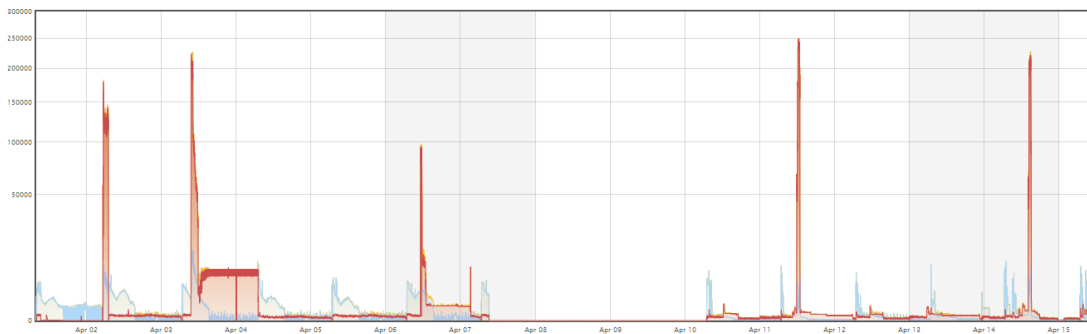
After the selection of an interesting time window, the view representing the internal network can be updated accordingly. *VACS* also includes metadata with information

---

<sup>2</sup> <https://www.elastic.co/>

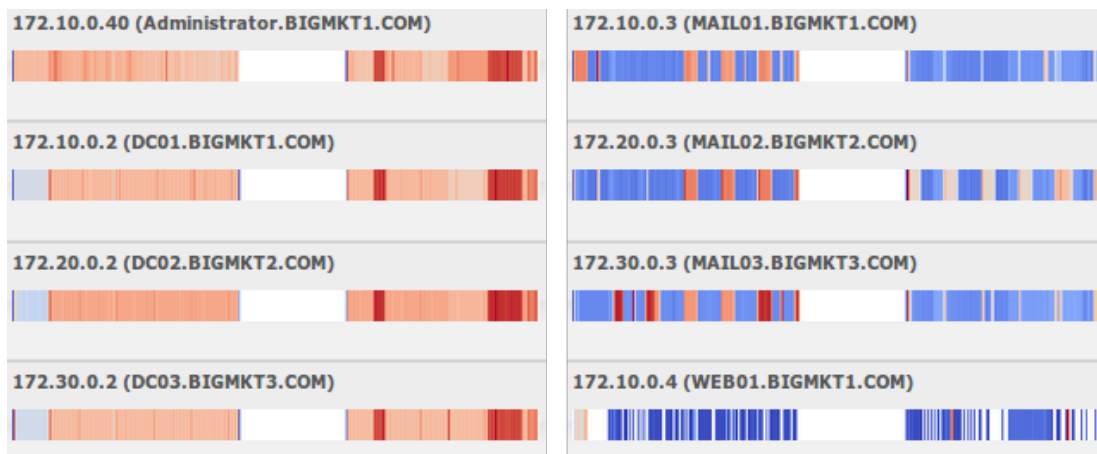


▲ **Figure 3.2** — Dashboard example in *VACS*. Some widgets on a dashboard showing the current situation using bullet graphs and a temporal histogram.



▲ **Figure 3.3** — Example of interactive line charts in *VACS*. Interactive line charts show the overall incoming and outgoing network traffic (number of flows). Different normalizations help to focus on peaks or low-traffic periods. Five enormous peaks are standing out in this example.

about the various network hosts within the internal network. This helps to distinguish between the various host classes (e.g., workstations, web servers, mail servers). To show many internal computer hosts and correlate their behavior within the selected time window, we employ a glyph-based technique to visualize the time-series for each host using a compact representation with colored stripes (Figure 3.4). By default, we use a blue to red colormap, in which dark blue colors refer to very low network activity (or any other selected measure) and red to an high amount of network activity. White areas within the compact representation, as seen in the middle of all glyphs in Figure 3.4,



▲ **Figure 3.4** — Example of striped thumbnail glyphs in *VACS*. Small multiple visualization of time-series metrics for internal hosts using striped thumbnail glyphs.

represent no data, which is important to identify network outages. The temporal glyphs for the internal hosts can also be seen on the left part in Figure 3.5, in which *VACS* is used on a large powerwall display to explore related external connections of selected internal hosts.

### Exploration of External Network Connections

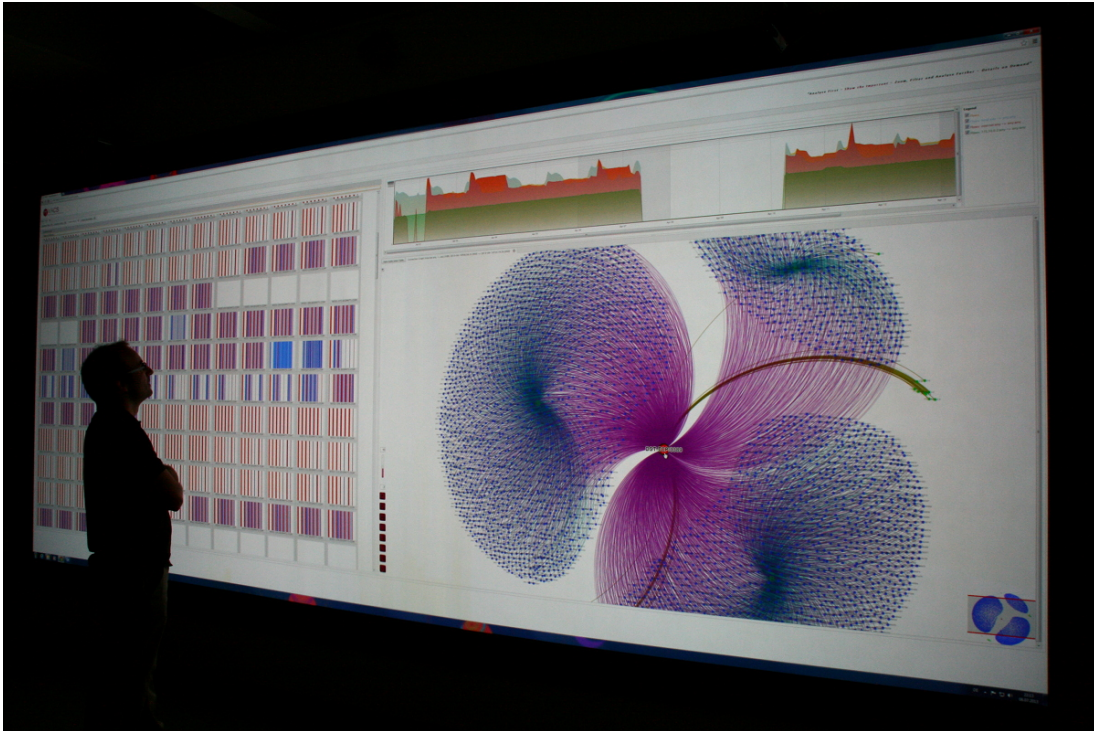
After selection of time windows and internal hosts, *VACS* can retrieve the actual network flow connections to various external ports and hosts from the database cluster. Figure 3.6 shows an example of the different connections between various hosts and port communications. This makes sense for shorter time spans or for specific queries, because it would be too cluttered to get an overall overview, however when the filters are applied which is intended in the proposed workflow, the analysis using such node-link diagrams becomes feasible and is often the preferred visualization to explore inter-dependencies as we found out in our field experiment with leading security experts from an operational security response team in November 2013 [88], which is further discussed in Section 4.4.3. However, often the automatically calculated force-directed layout is not perfect, so the user is able to interactively modify the node-link representation. An interactive fisheye lens can also be used to explore cluttered areas. Color is mapped to the different object types (e.g., IP addresses, source ports, destinations ports).

To make use of a more scalable summary visualization, we also include a treemap representation, to identify mostly used ports or hosts with the most traffic in a selected time span as seen in Figure 3.7.

Eventually, a data exploration table can be loaded as additional view. This table is quite important for the analyst, to show and export the underlying raw data to other applications or to generate actionable reports out of it.

### Usage Scenario and Analysis Workflow

An analyst wants to explore the past and the current network situation, because of several reachability and connectivity issues in the company’s network. After getting a basic idea using the dashboard about the current situation, he is interested in analyzing

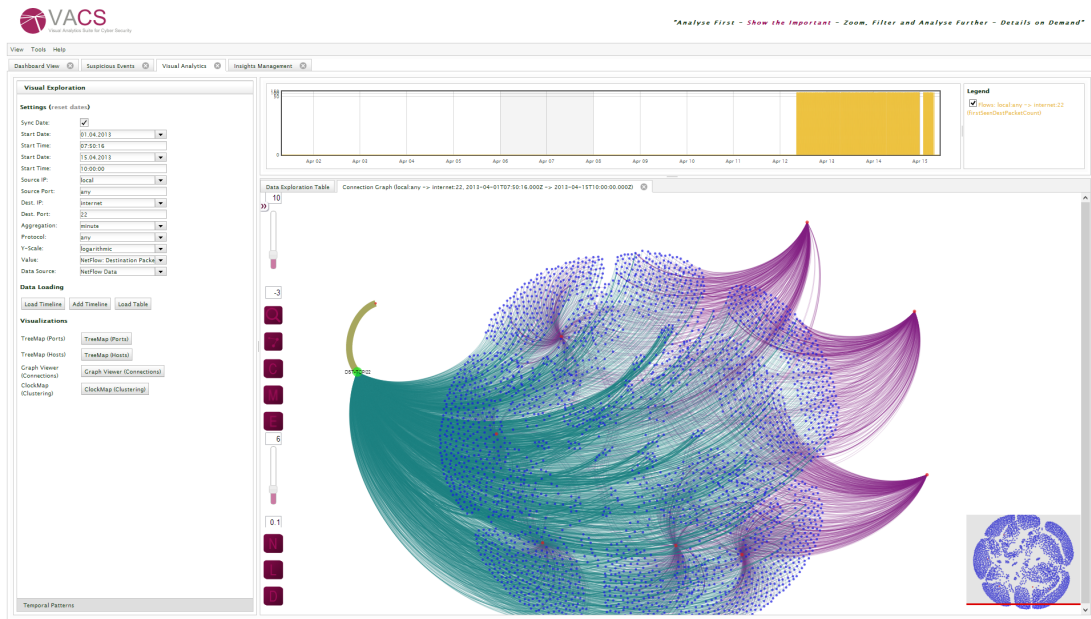


▲ **Figure 3.5** — *VACS* shown on a large powerwall display. *VACS* can be used on a large powerwall display. Several time-series are shown as interactive line charts on the top to select the overall time window. The colored striped thumbnail glyphs on the left represent the different traffic patterns for relevant internal network hosts. After selection of internal hosts, the interactive node-link diagram displays aggregated connections between different source and destination ports or other external hosts.

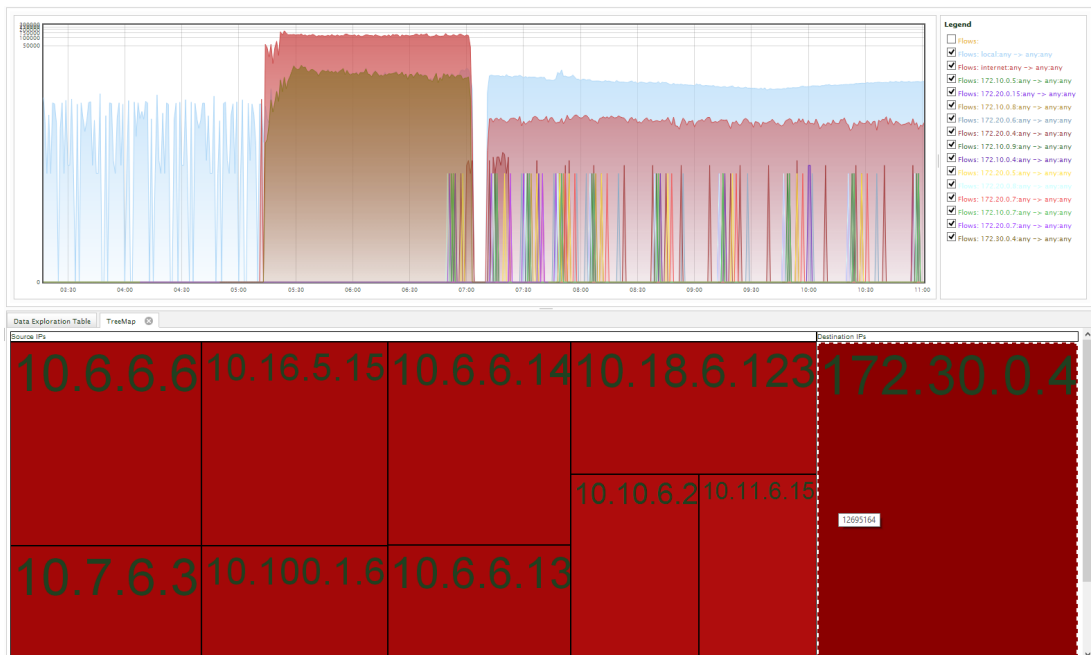
the overall incoming and outgoing network traffic. In this analysis he is especially interested to explore the reasons for the connectivity issues which happened several times. In Figure 3.3 he can clearly identify major traffic peaks on five different points in time. They seem to have slightly different patterns and differ in duration, extend and volume. The analyst can answer more questions by selecting the different high peaks using a rectangular selection. Additionally, he can add more related time-series to the line chart (e.g., different ports, different critical servers, number of alerts). Loading the treemap visualization even shows which hosts (or ports) are mostly involved and, therefore, responsible for the selected peak. With the help of this visual exploration possibilities he can distinguish between wide-spread denial-of-service (DoS) attacks or very specific attacks on various ports or just an ongoing company campaign with many legitimate connections. Further suspicious hosts can also be identified using the thumbnail glyphs in Figure 3.4 while the node-link diagram in Figure 3.6 helps to explore the aggregated connections of different hosts and attacks.

Overall, *VACS* can be used to analyze network activity with respect to internal and external hosts. The focus on internal hosts is given through the striped thumbnail glyphs to explore time-series in a compact way, while the connections during a selected timespan can be explored using an interactive node-link diagram.





▲ **Figure 3.6** — Example of an interactive node-link diagram in VACS. An interactive node-link diagram helps to analyze the aggregated connections between hosts and ports.



▲ **Figure 3.7** — Treemap overview of involved hosts in VACS. A treemap is loaded with underlying data from the selected time span. This helps to identify the top talkers (e.g., IP address with most activity in that time) or to get an overview of involved ports.

### 3.1.2 Evaluation using VAST Challenge 2013<sup>3</sup>

Evaluating a system like VACS is challenging, because many design decisions are involved, and evaluation of a complete systems is more than studying individual aspects in lab experiments. While best practices from visualization and design can be utilized, we also had a close collaboration with security experts within the VIS-SENSE [258] project, to gather feedback on early prototypes of the system.

However, to evaluate and compete with other international teams and to compare findings and acquired insights, we participated in the VAST Challenge 2013. The VAST Challenge 2013 consisted of three distinct mini-challenges. “*Mini-Challenge 1 (MC1) asked participants to use visual analytics to predict the success of new movies. Mini-Challenge 2 (MC2) focused on the design of a situation awareness display for monitoring the health, performance, and security of a large computer network. Mini-Challenge 3 (MC3) requested participants to identify the timeline of important network events in two weeks of network data for a fictitious marketing company*” [269].

In the context of cyber security, we participated in Mini-Challenge 2 and 3. For our MC2 submission about adaptive use-aware dashboard design [87], we received the award: “*Honorable Mention - Interesting Visualization Technique*” [269]. However, our primary focus was VACS which we used to address MC3 and achieved an “*Honorable Mention for Intriguing Visualization*” [269]. Furthermore, we also received valuable feedback from seven anonymous reviewers, involving experts from the field of cyber security and visualization.

#### Submission and Review Process

The VAST Challenge committee provided all datasets and various tasks and questions, which the participants needed to address. It was open to the participants which tools to use to solve the questions. Most teams started with state-of-the-art analysis tools. However, because of the limitations and missing capabilities, participating teams quickly moved on to specialized self-developed systems to address the challenge. The final submission consists of a written HTML page answering the questions using text and visualization screenshots. The participants were required to meet specific constraints in terms of word count and number of images to be used in their answer. This helped to have comparable submissions in the end. Furthermore, a 5 minute video with voice narration was required to show the workflow and interactive usage of the involved tools.

“*The VAST Challenge committee recruited reviewers with expertise either in visual analytics, information analysis, or application domains. (...) Subject matter experts were recruited from the pool of previous VAST Challenge reviewers and their social networks. A total of 66 reviewers participated, each providing from 1 to 10 reviews. Each submission received 4 to 7 anonymous peer reviews. All reviewers were given the opportunity to recommend entries for award consideration. Peer review questions varied across the individual challenges. However, in all cases, reviewers provided both ratings and explanatory comments*” [269].

Because of the complexity of the datasets “*accuracy reviews were performed by a small subcommittee of people very familiar with the data. These accuracy reviews identified the degree to which the submissions identified the events embedded in the data, but the accuracy reviewers also gave credit to submissions that identified other valid events in the data that were not intentionally embedded as part of the scenario*” [269].

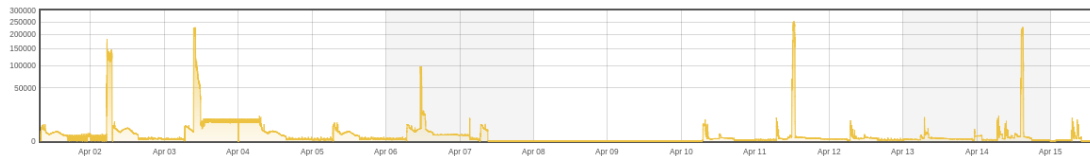
<sup>3</sup> <http://vacommunity.org/VAST+Challenge+2013>

## Background and Dataset Description

The challenge scenario was build around a story, about a fictitious company, called “*Big Marketing*”, which is an “*international marketing company employing a large staff of marketing executives who create and manage advertising and public relations campaigns for clients. Big Marketing has an internet research staff that stays current on the latest business, consumer and entertainment trends, searches for new markets, and comes up with ways to make Big Marketing’s clients stand out from the crowd. In addition, Big Marketing operates web sites for selected clients*” [210]. The participants should take the role as “*computer network manager, ensuring that Big Marketing networks are up and running for both the Internet-facing web services and the internal workforce. This responsibility encompasses the full range of maintaining current operations, planning for future needs, and securing and defending network assets against threats*” [210].

The provided data spans over a period of two weeks and consists of four major data sources: (i) meta data and network description, (ii) network flow data, (iii) network health and status reports, (iv) IPS alerts. Furthermore the participants could actively ask questions to the challenge committee during the weeks before the deadline.

As discussed in the introduction of Section 3.1 most tools for internal and external monitoring focus on packet traces and flow data, therefore, we also primarily used the provided NetFlow records as data source. Each record consists of 19 dimensions. The time period spans from 2013-04-01 07:30:00 until 2013-04-15 10:00:00 resulting in about 69,396,995 records (about 14 GB of raw data indexed in ElasticSearch) as seen in Figure 3.8. Further data analysis show that regular and legitimate network traffic was dominated by web browsing of staff members, customers accessing the web servers, e-mail traffic, and some FTP file transfers.



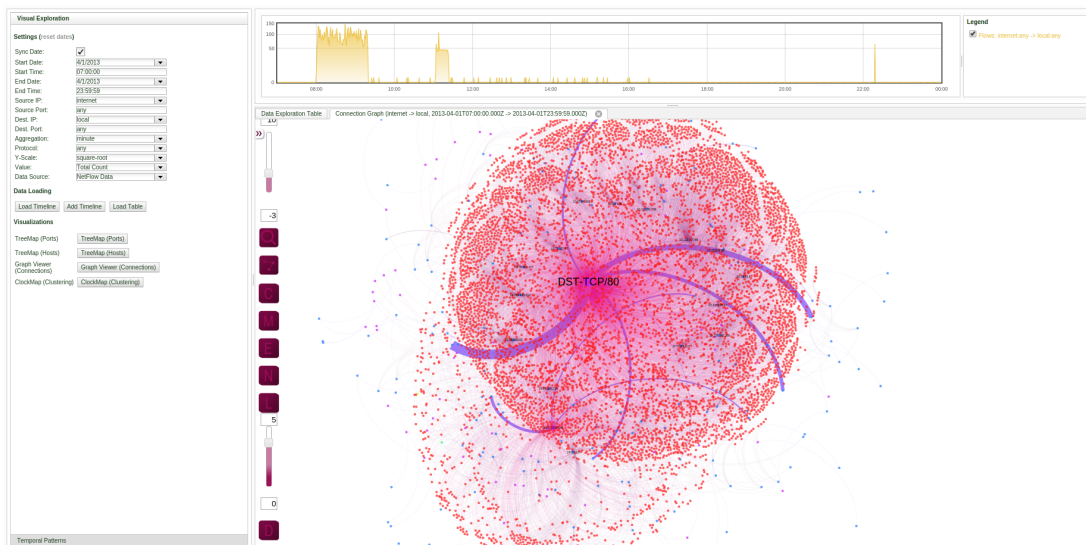
▲ **Figure 3.8** — Temporal overview of VAST Challenge’s network flow dataset. Overview of network flows as line chart using square-root normalization for the whole time period. This reveals huge data peaks, which make the analysis of subtle signals challenging.

The data description revealed that the network “*consists of three separate sites, each with its own domain controller, email server, web servers, and user workstations. The network is outfitted with a network flow collector which captures all of the traffic between Big Marketing and the (fictitious) internet used in this challenge, as well as a small portion of the internal Big Marketing traffic. In Week 2 of the data, the network is augmented with an Intrusion Protection System as well. (...) The Big Marketing web sites use addresses in the 172.x.x.x space internally. The internet in this scenario uses IPs in the 10.x.x.x address space*” [210].

### Questions for Mini-Challenge 3

The following questions and tasks are taken from the official submission entry form, which can be downloaded as package from the VAST Challenge 2013 website<sup>4</sup> or from the official “*Visual Analytics Benchmark Repository*” [210].

- Q1 “Provide a timeline (i.e., events organized in chronological order) of the notable events that occur in Big Marketing’s computer networks for the supplied data. Use all data at your disposal to identify up to twelve events and describe them to the extent possible. Your answer should be no more than 1000 words long and may contain up to twelve images” [210].
- Q2 “Speculate on one or more narratives that describe the events on the network. Provide a list of analytic hypotheses and/or unanswered questions about the notable events. In other words, if you were to hand off your timeline to an analyst who will conduct further investigation, what confirmations and/or answers would you like to see in their report back to you? Your answer should be no more than 300 words long and may contain up to three additional images” [210].
- Q3 “Describe the role that your visual analytics played in enabling discovery of the notable events (...). Describe whether your visual analytics play a role in formulating the questions (...). Your answer should be no more than 300 words long and may contain up to three additional images” [210].



▲ **Figure 3.9** — Overview of incoming network connections in VACS. As seen in the timeline at the top, almost all network traffic on 2013-04-01 happened in the morning, while there are few interesting peaks throughout the day. The node-link diagram reveals that the majority of the traffic relates to port TCP/80, referring to web browsing by staff members and web server responses to customers.

<sup>4</sup> <http://vacommunity.org/VAST+Challenge+2013>

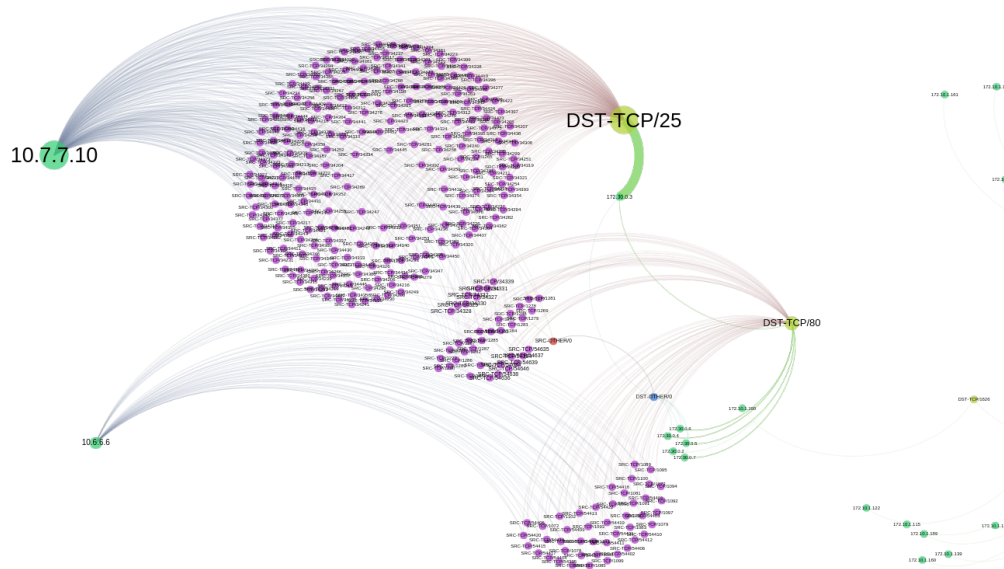
<sup>5</sup> The company takes the network down to investigate security concerns and to install an intrusion prevention system (IPS).

▼ **Table 3.2 — Evaluation of VACS using VAST Challenge 2013 MC3.** The ground truth of VAST Challenge 2013 MC3 consists of 29 official events and various bonus events. This table provides an overview about the verified findings. Found events in the competition and approved by reviewers are marked with ✓, while found events during exploration after the submission deadline and not reported in submission are marked with (✓). The reference value indicates the total percentage of submission entries identifying the event as provided by the committee.

Event ID	Subtlety	Event Type	Data Source	Reference	VACS	Figure
(1)	Questions only	Videoconference	-	0.0%	-	-
(2)	Questions only	Threatening Letter	-	0.0%	-	-
(3)	Subtle	Port Scans	NetFlow/BB	0.0%	(✓)	Fig. 3.10
(4)	Subtle	Port Scans	NetFlow	0.0%	(✓)	Fig. 3.10
(5)	Obvious	DoS	NetFlow	45.5%	✓	Fig. 3.11
(6a)	Subtle	Server Crash	NetFlow/BB	0.0%	×	-
(6b)	Subtle	Server Return	NetFlow	0.0%	×	-
(7)	Subtle	Port Scans	NetFlow	0.0%	×	-
(8a)	Obvious	DoS	NetFlow/BB	63.6%	✓	Fig. 3.12
(8b)	Obvious	DoS	NetFlow	63.6%	✓	Fig. 3.12
(9a)	Subtle	Server Crash	NetFlow/BB	36.4%	×	-
(9b)	Subtle	Server Return	NetFlow	36.4%	×	-
(10)	Subtle	Malicious Redirects	NetFlow	0.0%	×	-
(11)	Obvious	Exfiltration	NetFlow	18.2%	×	-
(12)	Obvious	Port Scans	NetFlow	40.9%	(✓)	Fig. 3.13
(13)	Obvious	Port Scans	NetFlow	22.7%	(✓)	Fig. 3.13
(14)	Obvious	Exfiltration	NetFlow	18.2%	×	-
(15)	Questions only	Threatening Letter	-	0.0%	-	-
(16)	Obvious	Network Down <sup>5</sup>	NetFlow	31.8%	✓	Fig. 3.4
(17)	Obvious	Port Scans	NetFlow/IPS	27.3%	×	-
(18)	Obvious	Port Scans	NetFlow/IPS	22.7%	×	-
(19)	Obvious	Failed DoS	NetFlow/IPS	36.4%	×	-
(20)	Obvious	Failed Exfiltration	IPS	18.2%	-	-
(21)	Obvious	Port Scans	NetFlow/IPS	18.2%	×	-
(22)	Subtle	Botnet Infection	NetFlow	9.1%	×	-
(23)	Obvious	Botnet Communication	NetFlow	36.4%	✓	Fig. 3.14
(24)	Obvious	Port Scans	NetFlow/IPS	9.1%	×	-
(25)	Obvious	Port Scans	NetFlow/IPS	18.2%	×	-
(26)	Obvious	Botnet DoS Attacks	NetFlow/IPS	18.2%	×	-
(27)	Obvious	Botnet DoS Attacks	NetFlow/IPS	9.1%	×	-
(28)	Obvious	Port Scans	NetFlow/IPS	22.7%	✓	Fig. 3.15
(29)	Obvious	Port Scans	NetFlow/IPS	22.7%	(✓)	Fig. 3.16
(Bonus)	Subtle	RDP Attacks	NetFlow	36.4%	✓	Fig. 3.16
(Bonus)	Obvious	General Port Scans	NetFlow	36.4%	✓	Fig. 3.10

### Solving the Challenge using *VACS*<sup>6</sup>

The ground truth for MC3 consists of 29 official events and various bonus events. For better readability, we include details from the *fictitious*, but realistic, ground truth scenario [210] as well. This actual scenario helps to connect the various events to each other, which makes the relations of the following events easier to understand.

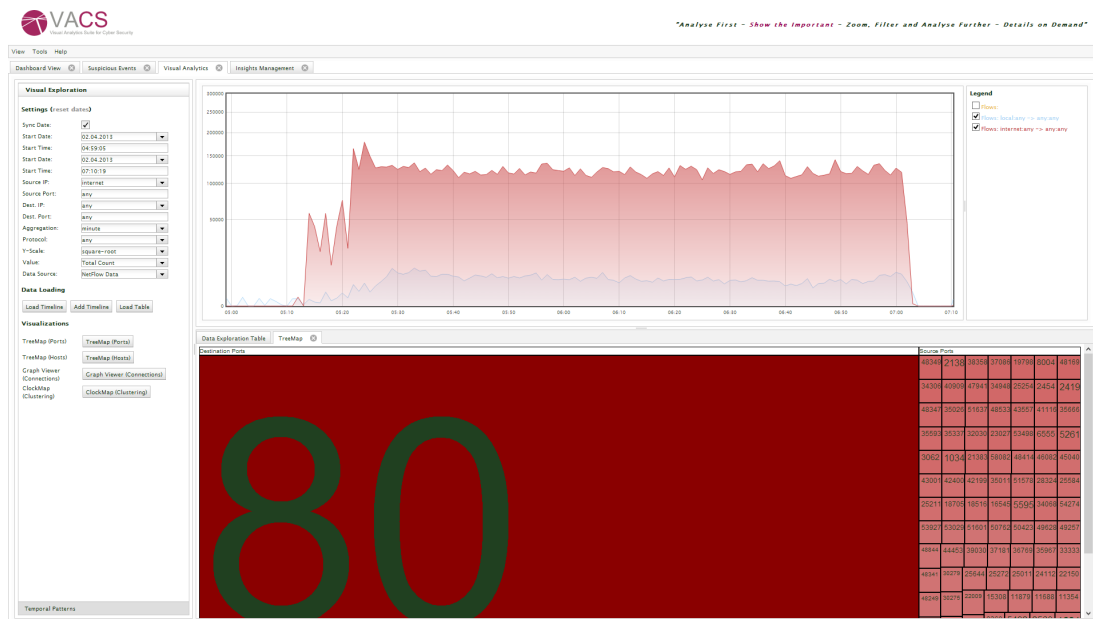


▲ **Figure 3.10** — **Interactive node-link diagram of port scans.** The interactive node-link diagram shows port scans with distinct patterns from 10.6.6.6 (Event 3) and 10.7.7.10 (Event 4) to primarily port TCP/80 and TCP/25. The visualization maps the source IP address (10.7.7.10) to a green circle, the lines to many magenta circles refer to the used source port addresses, the lines from those source ports all go through the light-green destination port (TCP/25) to various targeted web servers. The layout is calculated using a force-directed graph layout.

Table 3.2 provides an overview about the verified findings using *VACS*. While many events could be identified, there are even more, especially the subtle events, which were hard to catch. This actually highlights the need and more active usage of visualizations specialized for host monitoring to catch such events as well. One example, of such a technique is *ClockMap* [82], which will be discussed in Section 3.3. *ClockMap* was actually integrated in *VACS* but was not heavily used in our submission for the VAST Challenge 2013, because we decided to focus on the effectiveness of the other techniques.

Over the span of two weeks various anomalous network activities can be observed, in which a group called “*Butterfly Warriors*” is attacking Big Marketing over the two week period. Big Marketing is helping “*Total Crop Protection Services*” roll out a marketing campaign for “*Butterfly 2.0*”, an altered butterfly that will eventually lead to the extinction of natural butterflies. Prior to the dates covered by the dataset, the

<sup>6</sup> In this section, I use facts and descriptions from the the official ground truth of VAST Challenge 2013 Mini-Challenge 3 by Whiting et al. [269], which is available from the Visual Analytics Benchmark Repository [210] under Benchmarks / VAST Challenge 2013 / MC3 - Big Marketing / Solution.



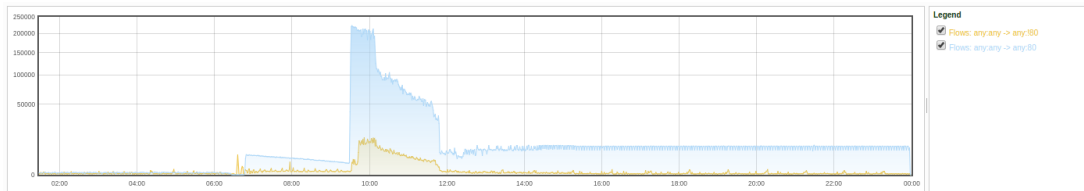
▲ **Figure 3.11** — Treemap representation to analyze DoS traffic. A denial of service (DoS) attack on 2013-04-02 between 05:10 and 07:10 (Event 5). The treemap obviously indicates that most traffic relates to port TCP/80 (HTTP traffic).

Butterfly Warriors send a threatening letter to Big Marketing, which refers to Event (1) and (2) in Table 3.2. As mentioned, challenge participants had the possibility to ask specific questions to the organizers. The letter would have been only provided, if someone would have asked for it. The reference value, for example, for Event (2) in Table 3.2 of 0% indicate, that actually no participant made a hypothesis about such possible threatening letter, hence no one could identify this event, which was also not provided in the other data sources.

Figure 3.9 reveals connections on 2013-04-01. As seen in the timeline at the top, almost all network traffic happens in the morning, while there are few interesting peaks throughout the day. The node-link diagram reveals that the majority of the traffic relates to port TCP/80, referring to web browsing by staff members and web server responses to customers. However, various individual IP addresses stand out in the node link diagram during interactive investigations. Restricting the time span and therefore filtering the view to the peaks throughout the day, reveals Figure 3.10, in which subtle port scans with distinct patterns from 10.6.6.6 and 10.7.7.10 to primarily port TCP/80 and TCP/25 can be easily detected. The visualization maps the source IP address (10.7.7.10) to a green circle, the lines to many magenta circles refer to the used source port addresses, the lines from those source ports all go through the light-green destination port (TCP/25) to various targeted web servers. These findings refer to Event (3) and (4) in which attackers of the Butterfly Warriors seem to investigate the network remotely.

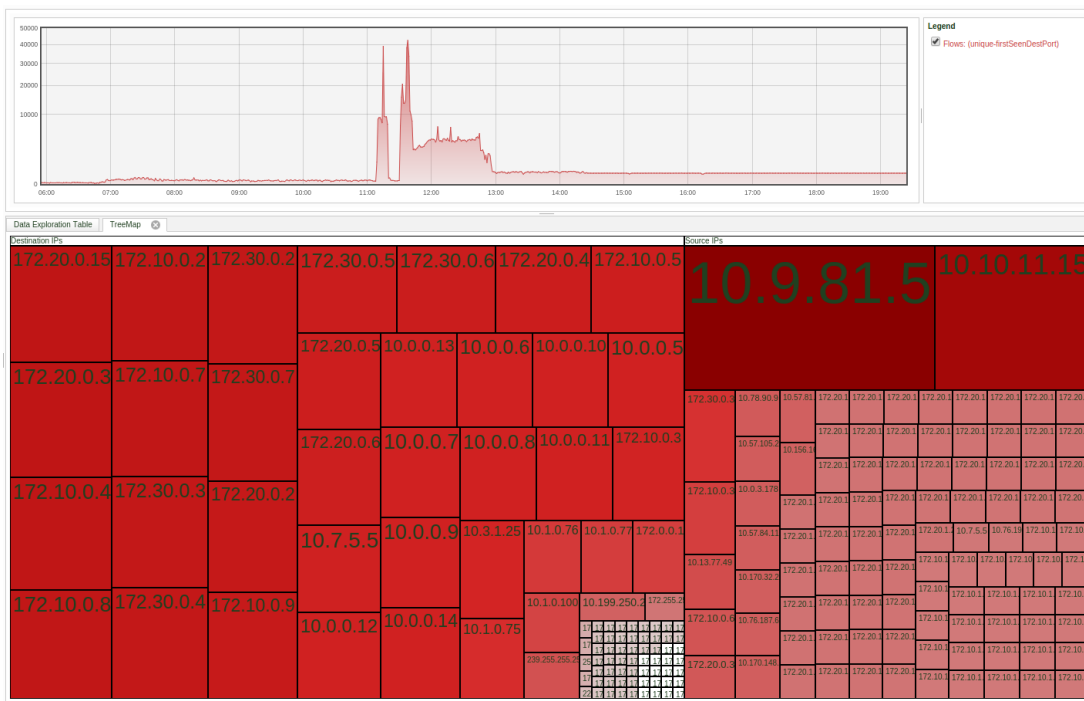
Using VACS a major denial-of-service (DoS) attack can be recognized (Event 5) on 2015-04-02 between 05:10 and 07:10 as seen in Figure 3.11. Obviously, the attackers started to heavily attack the network infrastructure. According to the ground truth, these attacks cause one server to crash (Event 6). Another server, which uses load

balancing across two servers, is able to withstand the attack and does not crash. However, these events and also the following subtle port scan (Event 7) could not be detected by any challenge participants. The DoS attacks continue on successive days. For example, on 2013-04-03 between 09:30 and 12:00 (Event 8) as presented in Figure 3.12.



▲ **Figure 3.12 — Timeline view for a DoS attack.** A denial-of-service (DoS) attack on 2013-04-03 between 09:30 and 12:00. The blue line represents traffic on TCP/80, while the yellow line represents traffic on all other ports. This highlights that most traffic involves TCP/80 traffic.

According to the ground truth scenario [210], the Butterfly Warriors also implant malicious code on one of the Big Marketing externally-facing websites, which did not leave any traces in the available data. Visitors to this affected website were immediately redirected to a malicious web server, where they were also infected with malicious code. The clues for this infection were very subtle. The only visible change was the decrease of session durations for visitors on the infected website (Event 10).

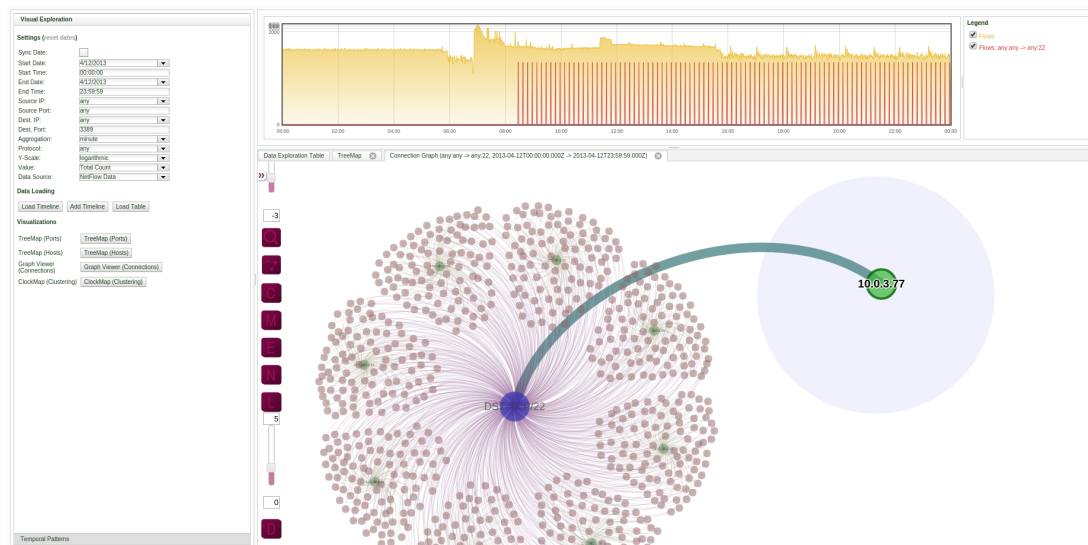


▲ **Figure 3.13 — Using the treemap for root cause identification.** Two major port scans from 10.9.81.5 and 10.10.11.15 on 2013-04-06 between 11:10 and 12:00 (Event 12 and 13). The treemap helps to identify the attack origins.



Because one of the infected computers belong to the system administrator for Big Marketing, the Butterfly Warriors use this vulnerability to open up all of the protected ports on the network. Various major port scans (Event 12 and 13) appear on 2013-04-06 between 11:10 and 12:00 as seen in Figure 3.13. Additionally, the attackers exfiltrate a couple of high value files from the Big Marketing network (Event 11 and 14). The ground truth [210] reveals, that the exfiltrated files were a file containing Big Marketing’s private client information and a recording of a video conference (referring to Event 1) between Big Marketing and Total Crop Protection Services discussing the marketing plan and the likely consequences of Butterfly 2.0. In this time Big Marketing also receive another threatening letter (Event 15) by the attackers. Furthermore, the system administrator discovers that important files were being exfiltrated. Therefore, he decides to pull Big Marketing off the internet in order to investigate and add an intrusion protection system (IPS).

This results in a three day gap in the data collection (Event 16), which is visible in various visualizations in *VACS* as for example in Figure 3.4. Various attacks in the second week are stopped by the IPS and couldn’t affect the network (Event 17-21). However, the Butterfly Warriors decide to post Big Marketing’s exfiltrated customer information on the Internet. Account managers navigate to these external sites hosting the leaked customer data and became infected with botnet malware (Event 22).

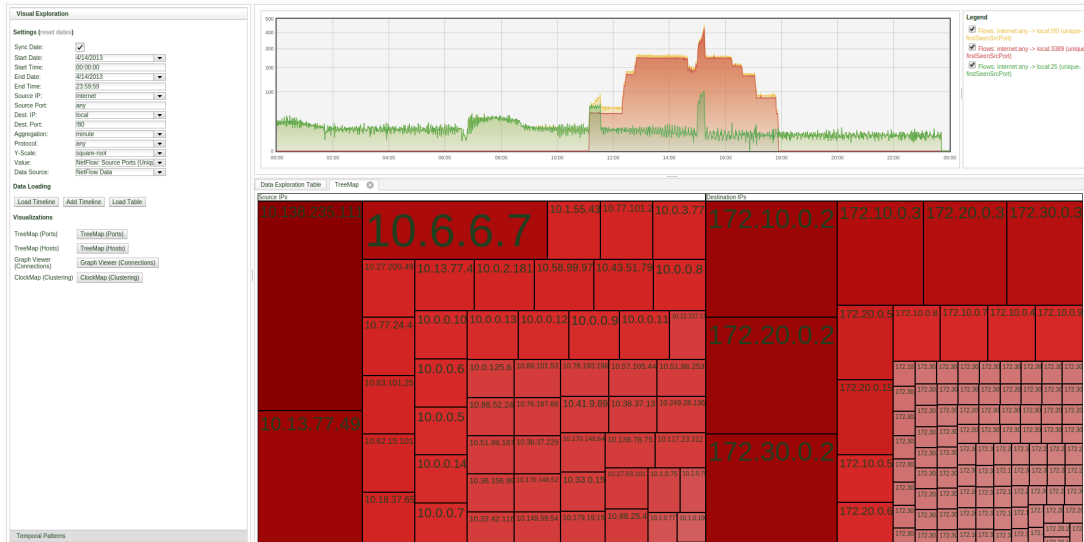


▲ **Figure 3.14** — Visualizing botnet communication. periodic botnet communication over SSH (TCP/22) starting at around 08:20 on 2013-04-12 (Event 23).

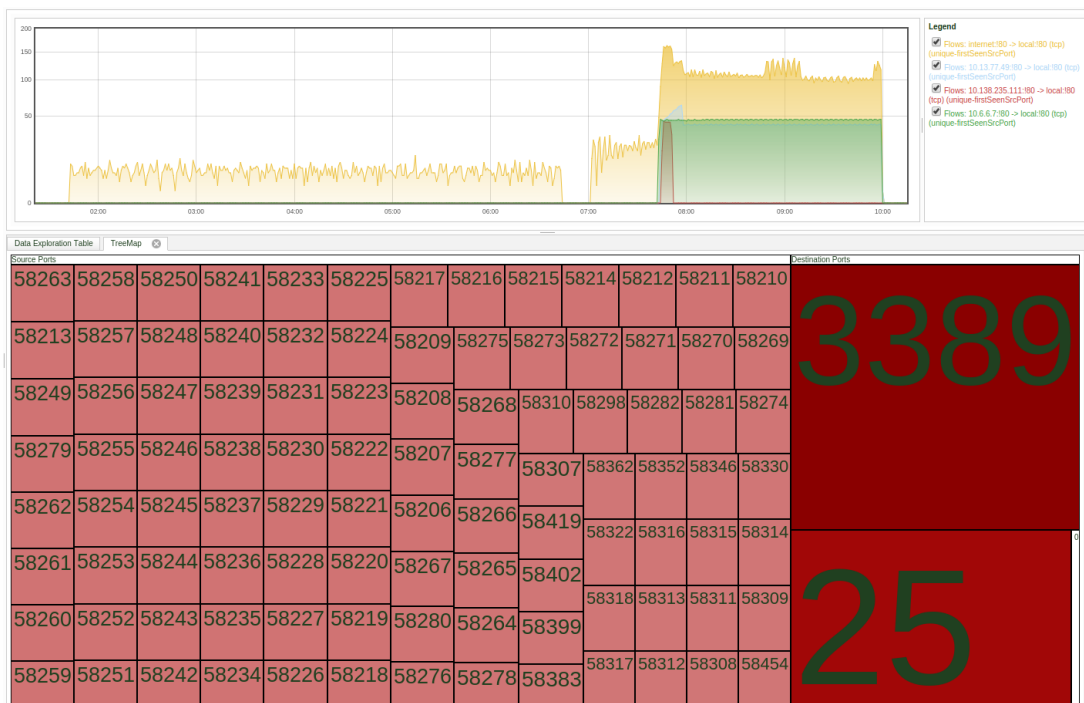
On 2013-04-12 08:20 the ongoing infection becomes visible, because infected machines start to communicate via SSH (Event 23) to a command and control server running at 10.0.3.77. Figure 3.14 highlights the regular and suddenly started SSH patterns in the timeline viewer at the top (red colored line) and visualizes the infected machines and the respective command and control server using a node-link diagram. Further port scans appear (Event 24 and 25) and 8 of the infected internal Big Marketing machines start a DoS attack against an external machine (Event 26 and 27).

Figure 3.15 shows various port scans from 10.13.77.49, 10.138.235.111, and 10.6.6.7 starting around 2013-04-14 12:20 (Event 28). Additionally, these attacks continue on

2013-04-15 07:45 (Event 29) primarily on port TCP/3389 and TCP/25, which is shown as dominating ports in the treemap as seen in Figure 3.16.



▲ **Figure 3.15** — Various sources of port scans on 2013-04-14. Identification of attackers orchestrating various port scans (Event 28) from 10.13.77.49, 10.138.235.111, and 10.6.6.7 starting around 12:20.



▲ **Figure 3.16** — Identification of most attacked ports on 2013-04-15. Various port scans on port TCP/3389 and TCP/25 from 10.13.77.49, 10.138.235.111, and 10.6.6.7 starting around 07:45 (Event 29).

### 3.1.3 Conclusions and Limitations

Overall the VAST Challenge 2013 Mini-Challenge 3 received 11 submissions [269]. None of the them were able to identify all ground truth events. However, with our submission using *VACS*, we were able to present and evaluate a scalable approach to analyze network flow data. Interestingly, we could identify many highly relevant events, even though we spend only a very limited amount of time in the actual analysis. Our web-based application uses a distributed database cluster to achieve horizontal scalability and combines state-of-the-art visual representations to assist the analyst in achieving situational awareness. We identified and provide means to explain unusual happenings in the network.

Overall, we received valuable and also very positive feedback from 7 reviewers and our approach was awarded with an honorable mention for a system with intriguing visualization capabilities. However, on the other side the active participation also helped us to identify various limitations and shortcomings.

- **Overplotting of superimposed line charts** – Line charts are the most common way to visualize time-series, which are easy to understand for the analysts. To solve the real-world tasks in the VAST Challenge it was often needed to compare and show several time-series at the same time. In *VACS* this is done by superimposing line graphs within the same chart. However, as for example also seen in Figure 3.7 this can become quite cluttered, because various colors are used to distinguish between the various time-series. Using small multiples as alternative is possible, but needs more screen estate. Furthermore, having the time-series near to each other as in superimposed line charts helps to recognize individual correlations better. Compared to pixel-based visualization, line graphs have the advantage, that they use position to encode the data value, which ensures quick recognition and judgment of the impact and amount of data.

*We use this limitation as one of the starting points for the technique proposed within IAS-Explorer in the following Section 3.2. The problem of correlating thousands of similar time-series can often be found in the use case to analyze port activity. Therefore, we propose a vertically aligned small multiple display to emphasize the comparison of individual segments of time-series data, to avoid overplotting and the usage of color encoding, which could have helped in network security scenarios as presented in VAST Challenge 2013.*

- **Lack of automated methods for correlation analysis** – Additionally, we missed to report various identified findings, because they were subtle and we misclassified them as regular network activity. Integrating analytical methods to support and enhance the visual representation would have the potential to address such issues. Highlighting interesting parts or automatic retrieval of related highly correlated time-series would have been beneficial.

*In VACS the analyst can use filters and search options to add various time-series to the visualization. However, often it would be helpful to get suggestions about highly correlated or similar time-series or temporal anomalies. For example, if the analyst identifies a peak in the time-series which represents flows/second over time, it would be very valuable if the system, would provide other more specific time-series for all individual TCP ports, in which this peak also appears. We also address this limitation in IAS-Explorer in the following Section 3.2.*

- **Missing contextual comparison for striped glyph thumbnails** – While the striped glyph thumbnails provide a compact way to represent and visually compare network activity of individual hosts, insights are limited, because a simple tabular layout doesn't provide enough overview for thousands of network hosts to spot outliers or anomalies. Visualizing them for example within the same IP subnet would help to compare individual hosts to the baseline of similar hosts in the same group to identify related hosts with similar patterns more easily.

*Similarly, in the use case of host and server monitoring, it is important to visually explore data within the context of other related hosts. We address this limitation using a novel visual representation called ClockMap which is scalable and compact enough to be used for visual exploration of network activity. We successfully integrate this technique into VACS and describe the details and outcomes in Section 3.3 and reevaluate the approach with the introduced VAST Challenge 2013.*

## 3.2 Visual Correlation for Port Activity Monitoring

The sections coming next mostly build on the following publication [226]<sup>7</sup>:

F. Stoffel, F. Fischer, and D. A. Keim. Finding Anomalies in Time-Series using Visual Correlation for Interactive Root Cause Analysis. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security, VizSec '13*, pages 65–72, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517966 [226].

In the following, we present *IAS-Explorer* focusing on techniques to analyze, visually explore, and monitor large numbers of network activity time-series. The main motivation for this work in the scope of this thesis is twofold: (i) supporting port activity use cases especially for privacy-preserving setups, and (ii) providing better means to correlate network activity time-series, addressing the limitations identified in Section 3.1.3.

Firstly, focusing on internal/external network monitoring comes with severe privacy issues. Analyzing network connections on a workstation-based level, which is easily possible with the techniques proposed in previous sections is a sensitive issue. On the one hand, it is important for network analysts, to identify potential misconfiguration, or even to detect compromised hosts to maintain the stability and security of computer networks. On the other hand, such visual exploration techniques might also lead to insights about specific user behavior resulting in ethical issues (which will briefly highlighted in Section 6.3). The observation to which servers an employee is connecting and what daily patterns a user might have – even unintentionally – can easily be seen by a system administrator utilizing such visual analytics tools. The moral and ethical issues of using such tools is not in the scope of this thesis, however, from a technical point

<sup>7</sup> The work was the result of the supervised master's thesis of Florian Stoffel. Based on previous discussions with network security experts, we had the idea to develop a visual analytics system for time-series correlation. I suggested to use scalable rotated time-series visualizations combined with similarity search to analyze the temporal network security dataset. We discussed the various steps together, while Florian did the implementation and introduced the analysis algorithms. Based on these results, we wrote a paper [226] together to introduce the visual analytics system. Daniel Keim gave advise and suggestions on the project.

of view, the focus on port activity using counter-based sensors instead of flow-based probes can be an alternative to privacy-invasive techniques.

For example, IAS [262] as presented by Waibel [262] and actively used by the Federal Office for Information Security (BSI) in Germany, doesn't capture any IP addresses. This internet analysis system extracts information based on descriptors from packet data and only increments and stores counters for them. This provides a privacy-preserving way to gather aggregated network activity data for large-scale computer networks. However, how should an analyst visually explore and correlate the time-series for millions of distinctive but often highly related descriptors?

Secondly, analyzing such temporal counters relate to the same issues as discussed in Section 3.1.3 in which the analyst extracts time-series from network flow data to visualize them as line charts. It is feasible to overlay several line graphs to compare the activity on various TCP ports over time. In *VACS* (Section 3.1.1), we superimpose various lines in the same visualization, which helps to directly compare and correlate them. However, when we select many time series the visualization gets cluttered and hard to correlate. For example, representing network activity for TCP/80 together with a few other well-known ports (e.g., TCP/443, TCP/25, TCP/22) is possible in such a visualization, but there are a total of 65,535 distinctive TCP port numbers, which are clearly not feasible to overlay. However, according to Fink et al. [76], it is very common for network analysts to utilize correlation in their daily work: “*Analysts perform standard types of correlation in the course of their normal work, such as correlating network flows to process activity*” [76]. In the same work, the authors quote analysts, that there is only very little visual support for such tasks [76]. Therefore, to address the challenges of visual correlation, we propose a juxtaposed approach using vertically arranged small multiples to represent the time-series and integrate analytics to show most interesting or highly correlated time-series.

## Related Work

The most extensive overview of visualization techniques for time-dependent data can be found in the book of Aigner et al. [5] providing a systematic overview and survey of many existing visualization techniques. A very compact visualization techniques is called two-tone pseudo coloring [204], which uses two discrete colors for each value of the time-series. This technique is also used and implemented in the so-called horizon charts, which properties have also been compared by Heer et al. [116] against line charts. However, using color to represent the value, restricts the further usage of color for highlighting critical or suspicious segments. Additionally, we are not so much interested in easy-to-detect peak values and the precise readability of the visual representation, which is a key advantage of horizon charts. It is more important to recognize shapes, correlations and patterns, where commonly used line charts provide a good basis.

The graphical perception for multiple time-series and line charts has been evaluated by Javed et al. [132], who showed that the presentation of time-series as small multiples is generally more efficient for comparisons across time-series with a *large* visual span. Plotting several lines in the same diagram was more efficient for comparison of *smaller* visual spans. This shows the trade-off we are confronted in our approach, because we are actually interested in large visual spans to convey the overall shape *and* small visual spans to correlate interesting anomalous segments against other time-series. *ChronoLens* [284] is a highly interactive approach which enhances the exploratory analysis of times-series. The user can select parts of the line charts. The data of the

selected segment is automatically transformed to show derivatives, correlations or other derived time-series for the selected focus lens area. This tightly integrates visual analysis with user interaction and provides good means to deeply analyze multiple line charts. With respect to the number of shown time-series, the *Line Graph Explorer* [139] is much more scalable, because it provides a compact overview using colored pixels positioned on a single line for each time-series. Selecting those pixel lines provides a lens mode to give more space to the selected metrics to be shown as standard line charts. This tool provides a compressed visual representation, which is very good to catch the overall global similarity of many time-series. However, if you need to explore many time-series in detail using the lens, the scalability degrades. While our technique is quite similar to the *Line Graph Explorer*, we have a stronger focus on comparing specific segments across thousands of metrics using the line chart to represent the relative value and using colored highlighting to emphasize the deviation to the underlying analytical model.

▼ **Table 3.3 — Related work for port activity monitoring.** Overview of related work with respect to data source and visualization type.

Method	Use Case			Data Source												Visualization	Year											
	Internal/External Monitoring	Port Activity Monitoring	Host/Server Monitoring	Packet Traces	Network Flows	IDS/IPS Alerts	Firewall Logs	Vulnerability Scans	Meta Data	System Metrics / Status Reports	DNS Logs	BGP Messages	Server Logs	File System Changes	Audit Trails	Webserver Logs	Database Logs	Honeypot Logs	Spam / Phishing Mails	Malware Files	Behavior Logs	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display	
PortVis [173]	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2004
Abdullah et al. [1]	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	2005
Existence Plots [131]	-	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2008
NetBytes Viewer [239]	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	2008
Mansmann et al. [168]	-	✓	-	-	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2012

Table 3.3 provides an overview for the most popular tools in the scope of port activity monitoring for cyber security. *PortVis* [173] also address our motivation concerning privacy issues in which data “*can only be coarsely detailed because of security concerns or other limitations*” [173]. McPherson et al. [173] specifically address the question: “*How can interesting security events still be discovered in data that lacks important details, such as IP addresses, network security alarms, and labels?*” [173] However, we focus on the visual correlation of arbitrary counters and not only for the given set of TCP/UDP ports. McPherson et al. [173] also primarily use dense pixel displays to represent the port activity using colored-matrix displays, while we stick to traditional line charts in a small multiple setting. Additionally, most of the other approaches do not integrate automated techniques to support the user, while our approach integrates correlation analysis and similarity search for time-series. Best et al. [24] do not specifically focus on port activity, but use advanced time-series analysis based on *Symbolic Aggregate approXimation* [138] to find unusual sequences to improve network security and to provide real-time situational awareness. Shafer et al. [211] also provide a visual analysis system for time-series monitoring to identify anomalies by decomposing significant

bursts and long-term trends. A good overview of related wavelet-based techniques to provide anomaly detection for security-related applications can be found in [135].

### 3.2.1 IAS-Explorer – Visual Analytics for Port Activity Correlation

To address the challenge of visually correlate vast amounts of time-series to support port activity monitoring, we build a system called *IAS-Explorer*. We decided to build a separate application focusing on this topic, however, the developed techniques could be integrated into *VACS* as well. The server-side component of *IAS-Explorer* is responsible for managing the time-series data and providing analysis, query and retrieval related functionality. A rich client application is then used to provide a graphical user interface to explore and analyze the data. Multiple clients can operate independently from each other with data from the same server instance.

The core of the server is a custom time-series database, which acts as the time-series persistence layer. Although designed as a high performance retrieval system for time-series data, the database also fills out missing values or re-samples the data with a given interval transparently. This leads to a consistent dataset without missing values, which allows the simplification of further analysis and the processing outside the server. In addition, the resulting time-series are continuous in the time domain, which is a requirement for the Fourier analysis. By default, the server provides support for similarity queries. By specifying an originating time-series or its model and a time-span, the server can search in a set of given candidates or the complete, available time-series stored locally. By default, the distance of two time-series is computed by the euclidean distance of the normalized query region. Thanks to the retrieval performance, the server can finish a time-series query on a dataset of around 1.1 million time-series in about a minute (10 months of data, indexed in five minutes intervals, Intel Core 2 Quad Processor, 8 GB of main memory, Intel X-18 SSD).

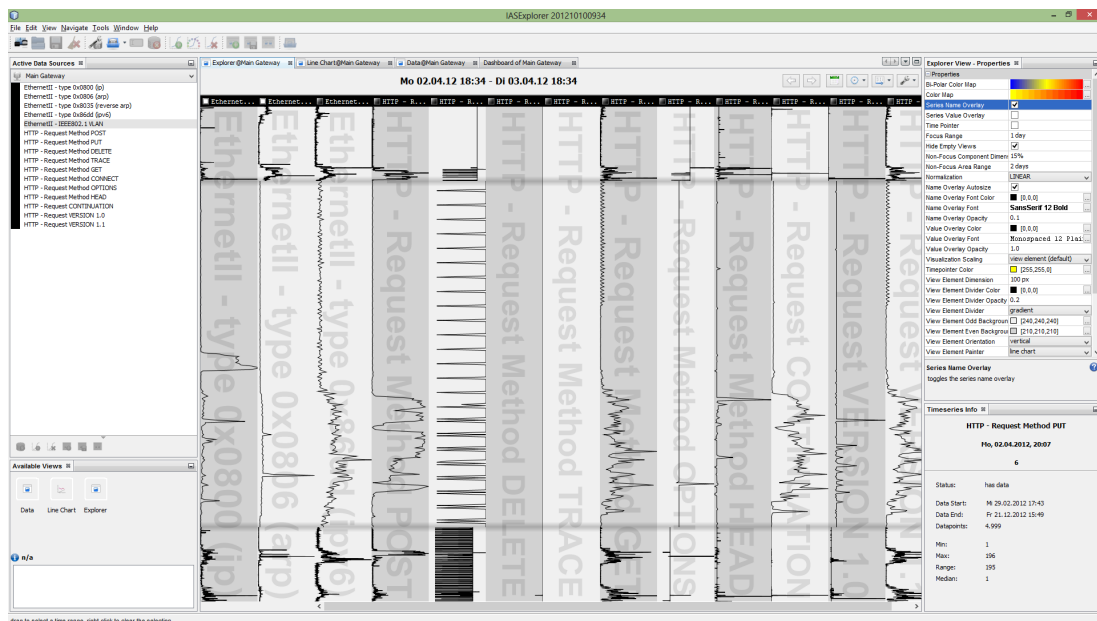
#### Time-Series Modeling

Besides the data restoration and sampling, a model of the time-series is created or updated when new data is inserted in the database. This model can be retrieved by the client and supports additional visualization and analysis methods.

In general, there are certain key observations characterizing a network time-series on two different levels. The first level is the intra-day level, where the observations refer to phenomena lasting a few hours. The second level where key observations can be made is the day level. A good example of such a day level key observation can be made when comparing the overall shape of a time-series of labor- with non-labor days.

Those observations are the motivation of creating the time-series model per day. Each time-series is modeled by seven independent models describing one weekday. There is no distinction in holidays or vacations, which preserves the maximal generality of the model on server side. Such adjustments should be made on client side, where in the ideal case the user can interactively adjust any kind of filters or modifications on the data. This also opens possibilities for task-specific adaptations of the model, where the server is just providing general data and the client adapts them in a task specific way.

The model for one time-series contains two different models created by Fourier and wavelet transform of the time-series [26, 190]. In general both methods can be used to analyze and model time-series data. The Fourier transform decomposes the signal in components, where each of the component can be interpreted as a longer or



▲ **Figure 3.17** — The main interface of *IAS-Explorer*. On the left side the data management and time-series selection window is shown. In the main area various visualizations can be shown (e.g., the *Explorer View*). On the right side the user can adjust settings to fully configure the visualizations. Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

shorter lasting phenomena in the time-series data. Besides this advantage, the frequency domain data resulting from the Fourier transform loses its time dimension. Therefore, it is almost impossible to properly model non-stationary signals which may change the frequency over time, or very short lasting phenomena in general. To overcome this limitation of the Fourier transform based models, an additional model based on the wavelet transform has been added. The major advantage of the wavelet transform is the dynamic window size, since the actual wavelet function is scaled to fit the input in data and time domain. Together, both parts of the model can accurately capture different longer lasting effects and also capture short phenomena in the time-series. To maintain the general nature of supported analysis tasks by the server and the models, there is no combination on the server side of the Fourier transform and the wavelet transform of the time-series, but band-filtering of the models is supported. By choosing such a design, the server does not restrict the available analysis tasks, but at the same time supports common, potentially computation intensive filter techniques. To create a Fourier and wavelet model out of different days, the resulting coefficients are aggregated incrementally [141]. Besides being able to compute the incremental arithmetic mean efficiently, a comparison of different aggregation methods has been made by creating models out of 9 weeks of real network time-series data. To judge the quality of the aggregation method, the resulting models have been evaluated with the sum of squared residuals of the models and the input time-series. The resulting model can be used to find anomalies by comparing the actual value of a time-series with its aggregated model. The server returns both, the Fourier and wavelet model, which keeps the design space of the application and its processing and application of the model as general as possible.

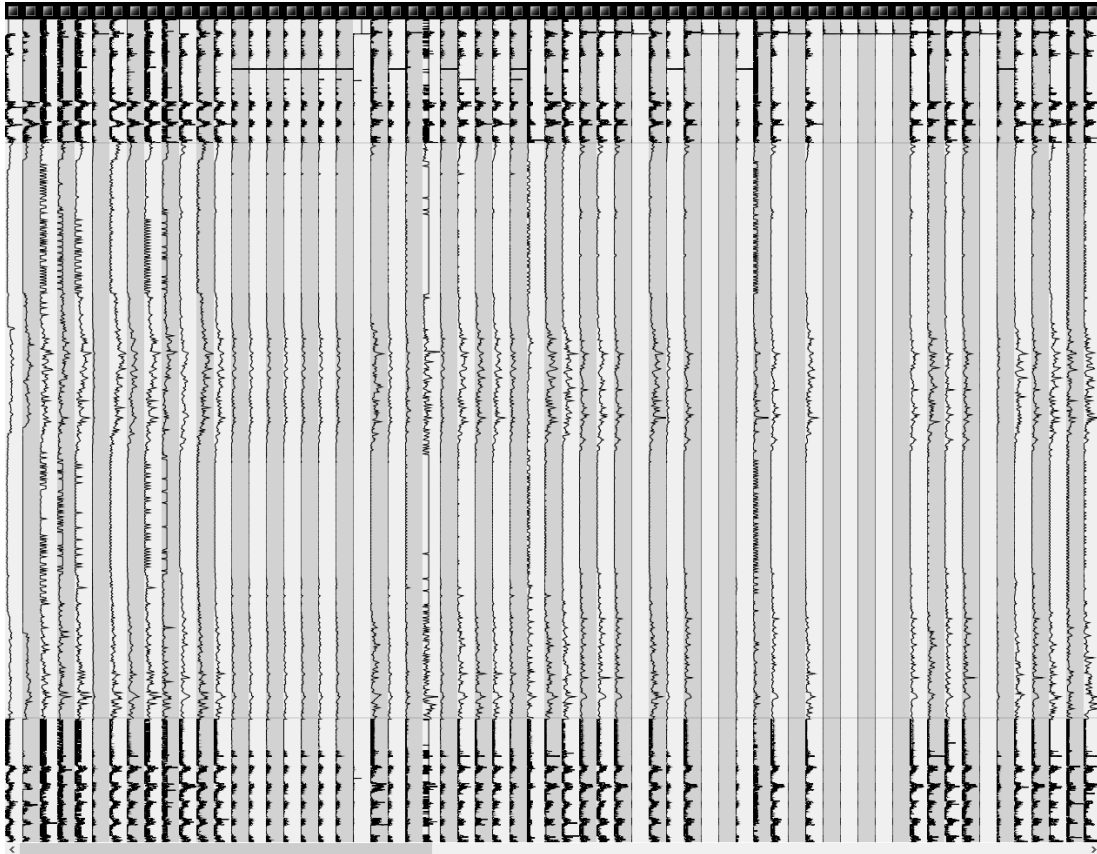


## Graphical User Interface

The overall interface of the client can be seen in Figure 3.17. It is composed out of three main areas. On the left, there is the general data control and action area. The center of the user interface, is designated to hold the visualizations (e.g., the *Explorer View*). On the right area, various settings can be adjusted and context sensitive displays are placed. This panel is also used to display details on demand, if the current context provides such detail information. For example, if segments of one or multiple time-series are selected, the minimum and maximum values of those selections will be shown. All parts of the user interface can be detached and freely placed on or even outside of the main application window. This also adds support for multi-monitor workplaces.

The *Explorer View* is built to support the following tasks: (i) *shape recognition*: similar time-series should have similar visual appearance and shape. (ii) *correlation recognition*: users should be able to visually identify time-series with high correlation, and (iii) *pattern recognition*: The visualization should enable the user to recognize similar patterns in different time-series. In addition, those tasks should also scale for large amount of time-series. Our visual approach takes the context of the time-series into account and allows refinements of the visual representation, which is desirable in order not to lose any information. Also, the time resolution of the *Explorer View* is freely adjustable and the visual appearance can be adjusted to fit the task best. The visual interface also allows exploration and browsing through the data, which should create a picture of the network condition and its usual patterns. In addition to fulfilling the task specific requirements, line chart based time-series visualizations have two further advantages. To label data in line charts is straight forward by re-using the usually empty area in the background of the chart. Besides having the possibility of enriching line charts with additional data, the scaling invariance of the actual line shape facilitates level independent shape, correlation and pattern recognition.

Due to the layered network architecture, this property is desirable because a network operation can have effects on different network time-series. For example, browsing to a website generates data in (not only) the following features shown as time-series: IP Traffic, TCP/80, and HTTP. Therefore, it is very likely that time-series, generated from the different layer data, are composed of parts of the same operations. Scaling the series in a fixed range, for example  $[0 \dots 1]$ , creates similar line charts in terms of their shape and correlation. Obviously, this also helps with the visual correlation and pattern recognition. The line charts in the *Explorer View* differ in one important aspect from common line charts. The time axis is not on the horizontal, but on the vertical. While this is not conform to the common line chart displays, it has an effect on the perception of the operator. In the Western world, people are used to read text and charts from the left to the right. This is also the case for line charts. This leads to the behavior, that viewers tend to follow a single line chart, instead of comparing them to each other even if there are multiple charts drawn next to each other. By placing the time axis not on the horizontal, but on the vertical, we force the viewer to break this habit, and try to direct the perception to comparing different line charts. The *Explorer View* does not force this rotated view, the single series displays can also be rotated by 90 degrees, which results in a common line chart arrangement. To account for the nature of repeating patterns in the data, it is desirable that the visualization is able to put the currently focused pattern in the larger context of the series. To support that, each line chart is divided into three parts: the prae-focus, focus, and post-focus area. The prae- and post-focus area are building the context area, the focus is located in the



▲ **Figure 3.18** — Explorer view showing 63 different time-series. Each series is scaled in a way, that the data of all time-series fit on a common workstation display. The order of the time-series plots is determined by the volatility of the data in the focus area. Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

middle of the visualization area. One key issue is the blending area of the non-focus with the focus area, which is caused by the different scaling of focus and context area. There are numerous different methods of techniques that those areas of different scales have a smooth transition to each other, for example based on a Gaussian kernel or hyperbolic functions [39]. In our case, comparison and exploration requires to have the current interesting points in the focus area of the visualization. To have a steady reminder of the different scales in terms of time and to minimize artifacts introduced by distortions introduced by the time scaling techniques, the *Explorer View* uses a sharp transition from the context to the focus area. An additional shadowing around the area transitions can be enabled, to make the actual borders of the three areas clear to the observer. This shadowing can be seen in Figure 3.17. Each time-series is displayed as a single line chart, which according to Javed et al. [132] is the right choice for the discrimination and therefore also the compare task. In the same work, the authors show that displaying time-series with less space has little influence on the time the analyst needs to accomplish a given task. The smaller size has only an effect on the ability of estimating the value of the time-series, which is not a key issue in the tasks the *Explorer View* is designed to support. Especially for tasks, where many time-series have to be considered at once, this property is important. To fit as many time-series on the

available display space as possible, all plots in one *Explorer View* instance can be freely resized to fit the needs of the task and visual abilities of the analyst. In Figure 3.18, a view with 63 different time-series is displayed. Although the space used to display the time-series charts is very small, it is possible to get an impression about their shapes and compare them with each other. Creating and executing queries on the displayed time-series is supported by the visualization. To issue a similarity query, the analyst can choose an area of a time-series via clicking and dragging the query time-span directly on the visualization. After selection the query range, it is possible to narrow down the search space and name the query before it is executed. The results can be displayed in any *Explorer View* instance and inspected visually.

### 3.2.2 Evaluation using Port Correlation Case Study

In this section, we describe how our system can be used to correlate many time-series within a computer network with around 20 users. Due to the general nature of network traffic, the definition of an *anomaly* can be different. In the following, we define an anomaly as a significant deviation from the usual traffic levels. The threshold of allowed deviation from the time-series to the model can be adjusted in multiples of variances of the time-series model.

For the following case study, the network traffic of a small computer network with a mixed environment of around 30 workstations and servers with about 20 regular users, has been analyzed on different network layers. To do so, a *probe* analyzes the traffic going through a central switch by trying to match *descriptors* to the data.

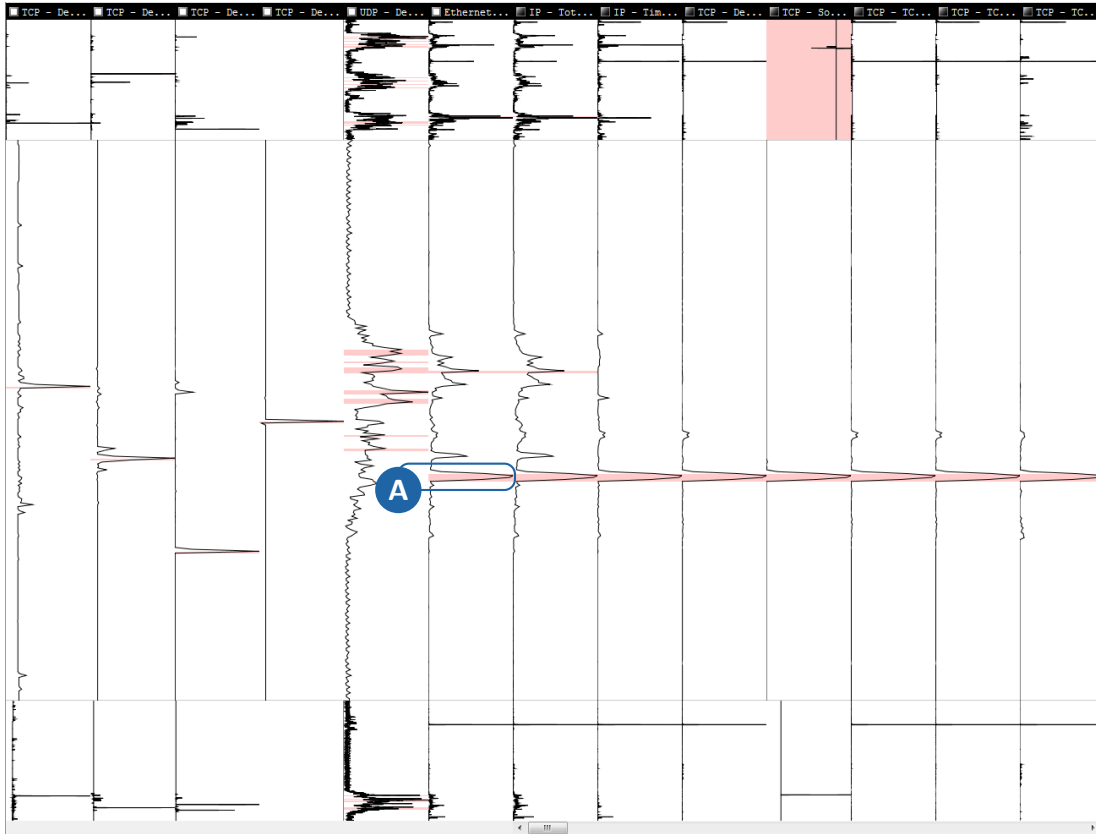
The analysis system contains descriptors for different protocols like TCP or UDP, SIP or HTTP, and application specifics, for example for each IRC command. For each of those, a numerical counter exists, which is incremented each time a descriptor matches. The counters are transmitted in five minute intervals to a data store, from which applications can retrieve the counter values and build a discrete time-series out of them. In the deployed system, a total of 1.6 million descriptors are contained, from which around 300,000 matched in the captured traffic of the observed network.

Since the dataset contains numerical counters only, sensitive data like source IP, destination IP, or the application payload can not be stored, which protects the privacy of the users. While it is possible to use this dataset for traffic and application usage analysis, it is not possible to conclude which workstations or servers are behaving

▼ **Table 3.4 — Overview of selected descriptors.** A time-series group containing some of the network time-series belonging to the most widely exploited services.

ID	Descriptor	Service	Description
#1	TCP - Destination port 25	TCP/25	simple mail transfer protocol (SMTP)
#2	TCP - Destination port 194	TCP/194	internet relay chat (IRC)
#3	TCP - Destination port 465	TCP/465	secure e-mail transport (SMTPS)
#4	TCP - Destination port 587	TCP/587	secure e-mail submission (SMTPS)
#5	TCP - Destination port 6667	TCP/6667	internet relay chat (IRC)
#6	UDP - Destination port 53	UDP/53	domain name resolution (DNS)
#7	EthernetII - type 0x0800 (ip)	IPv4	count for IPv4 packets

anomalous. To overcome this limitation, multiple probes can be added to different subnets or in front of single servers. Unfortunately, in our environment this was not possible due to user concerns regarding their privacy.



▲ **Figure 3.19** — Overview of time-series for various descriptors. The *Explorer View* showing the time-series selected at the beginning (the first six) and the time-series returned by the server by the similarity query (the last seven). Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

### Root Cause Analysis using Port Activity Correlation

Our use case begins with the analyst browsing through the network time-series data. Our system is capable of storing groups of time-series, so that if the active data source contains series with the given name, they can be loaded quickly. In our example, the analyst has created a time-series group containing the time-series shown in Table 3.4. This group contains time-series (descriptors) describing the most vulnerable services, which are usually target of attacks and are used to be exploited in various ways. Therefore, anomalies in those series require special attention, because they often a sign of unwanted network activity. The following interaction workflow is visually represented in Figure 3.20.

To support browsing through data, the *Explorer View* visualization is switched to the model difference mode, where significant deviations of a time-series from its model are highlighted with a blue (lower value as modeled) or red (higher value as modeled) background. This view mode is realized by querying the server for the time-series model,

applying the reverse transformations with the configured band-filters, computing the differences of the model and displaying them in the background of the line-chart.

▼ **Table 3.5 — Overview of automatically retrieved descriptors.** The first seven series returned by the server when the analyst queried for the anomaly region (A) he visually identified as seen in Figure 3.19.

ID	Descriptor	Service
#1	IP - Packet length between 0 and 255	IP
#2	IP - Packet TTL between 64 and 95	IP
#3	TCP - Destination Port 22	TCP/22
#4	TCP - Source Port 36761	TCP/36761
#5	TCP - Destination Port 22 and packet length 0 - 2557	TCP/22
#6	TCP - Window Size 4096 - 4351	TCP
#7	TCP - Window Size 3840 - 4095	TCP

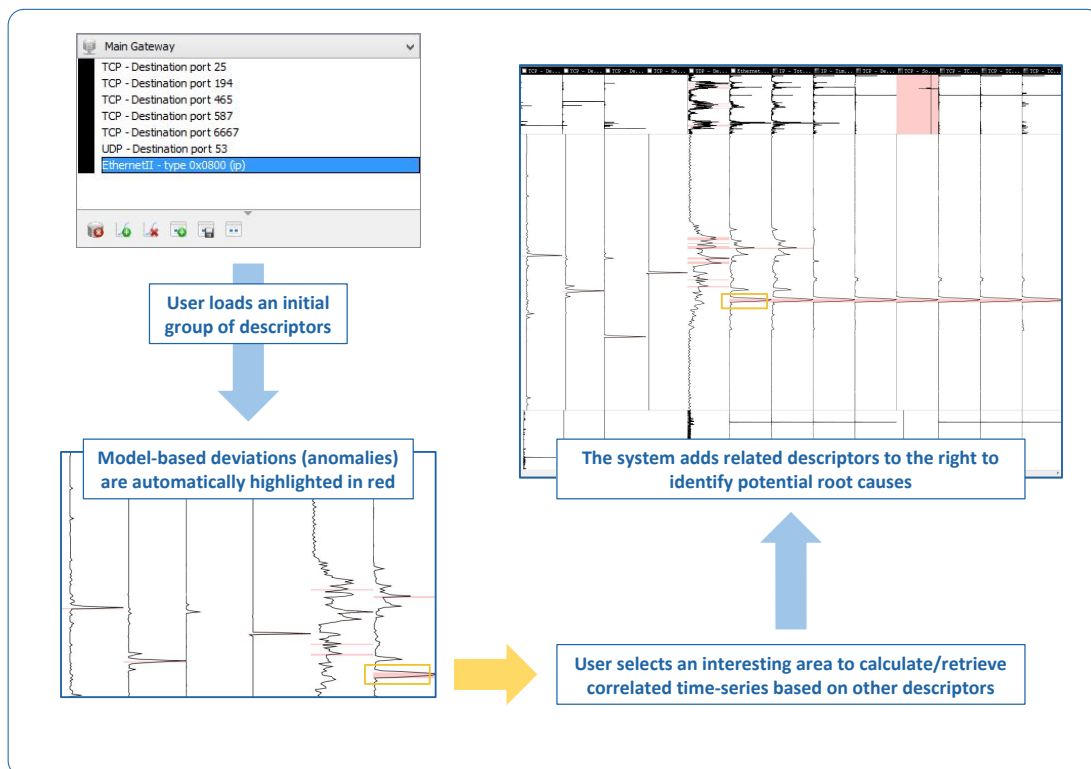
The browsing task can be performed by pressing the arrow keys or scrolling through the data with the mouse wheel. This simple interaction induces only low cognitive effort and allows the analyst to concentrate on the visual correlation of the time-series and on detecting anomalous areas via the background color of the visualization.

By browsing through the data over time, the analyst spots an area, where the general level of ethernet network traffic has a significant spike, which is identified as large deviation from the model as seen in step 2 of Figure 3.20. Selecting the range of the anomaly with the mouse and formulate a similarity query, which is executed on the server, is the next step towards identifying the cause of the traffic spike. After the query has finished, the analyst has the possibility of getting a list of resulting time-series ordered by their similarity, or adding them in the visualization for visual correlation analysis. Both, the visualization (Figure 3.19) and the list of similar time-series (Table 3.5) indicate a very large, unexpected transfer of data to hosts outside of the monitored network on port TCP/22. On the visualization side, the analyst can clearly see that the selected spike (A) of the ethernet time-series in Figure 3.19 is contained in all other visible time-series on the right of the originally queried series. By that, the analyst can conclude that there are some very good candidates to get an impression of the application and the actual root cause. This is strengthened by the fact, that there are also no anomalous spikes in the focus area of the visualization. This is additional information which can not be seen when just a list of similar time-series is returned by the server. For all displayed time-series, the spike detected in the aggregated network traffic is an anomaly which can be easily spotted by the operator. Having a look at those series (Table 3.5), it becomes clear that a large transfer of data has happened. The destination port TCP/22 is usually used for SSH based services, and there are some protocols which use SSH as transport protocol for their application data like *SFTP*<sup>8</sup> or *rsync*. Together with the detected anomaly in the aggregated network traffic, the analyst can conclude that most likely a large transfer of data from the internal network to a machine in the Internet has been executed relating to a possible data exfiltration.

<sup>8</sup> SFTP: SSH File Transfer Protocol

### 3.2.3 Conclusions and Limitations

To address privacy issues and support the analysis of many time-series, we built a visual analytics system for examining and investigating time-series data. It provides tight coupling of analytical models and visual representations capable of mining through vast amounts of time-series data. To support this task, the system features a focus plus context or lens based line chart carefully designed for displaying correlation of sub-segments of time-series. The usefulness of the design has been shown with a case study where the system allows an analyst to determine possible causes of a traffic anomaly. Currently, we implemented only simple analytic models as proof of concept, while the integration of more sophisticated analysis techniques would lead to even better results. The *Explorer View* is also limited and could be enhanced with further visual representations, for example based on glyphs designed specifically for showing anomalies in time-series data. In addition to the automatic ordering of the series, it is also desirable to identify groups and aggregate their visual representation in order to reduce the number of visualizations shown at once. Although preliminary tests and discussions had been promising, the *Explorer View* with its 90 degree rotation of the line charts is not yet formally evaluated in contrast to alternative visualization approaches.



▲ **Figure 3.20** — Interaction workflow of *IAS-Explorer*. This interactive process when using *IAS-Explorer* shows the combination of user interaction and analytical guidances by the underlying models.

### 3.3 Visual Exploration for Host and Server Monitoring

This section builds mostly on the following publications [82, 86]<sup>9</sup>:

F. Fischer, J. Fuchs, and F. Mansmann. ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data. In M. Meyer and T. Weinkauf, editors, *Proceedings of the Eurographics Conference on Visualization (EuroVis - Short Papers)*, pages 97–101, Vienna, Austria, 2012. The Eurographics Association. ISBN 978-3-905673-91-3. doi:10.2312/PE/EuroVisShort/EuroVisShort2012/097-101 [82].

F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. Visual Analytics zur Firewall-Konfiguration und Analyse von Netzwerkverkehr (in German). In B. f. S. i. d. Informationstechnik, editor, *Informationssicherheit stärken - Vertrauen in die Zukunft schaffen: Tagungsband zum 13. Deutschen IT-Sicherheitskongress (in German)*, pages 273–283. SecuMedia Verlag, 2013 [86].

In the following, we present *ClockMap* which is a novel technique specifically designed for host and server monitoring to visually explore time-series of large numbers of network hosts within a given hierarchical context. The main motivation for this work in the scope of this thesis is twofold: (i) supporting host and server monitoring use cases, and (ii) providing context-awareness for host and server monitoring, addressing the limitations identified in Section 3.1.3.

Especially for the analysis of network traffic of large computer networks, it is important to monitor the network usage to detect anomalies or to understand the behavior at different levels of detail. On the one hand, there is the need to gain an overview about the current situation. However, obtaining details and more information is crucial to understand such overall trends to eventually identify the underlying cause.

Furthermore, many real-world datasets contain an intrinsic hierarchy, which can provide important information to the analyst. However, analyzing related work as seen in Table 3.6, reveals that such hierarchical aspects are widely neglected in the utilized temporal visualization techniques.

In network security, for example, such a hierarchy is often given through the network definitions encoded in prefixes of IP addresses. Alternatively, computer networks can be structured according to organizational groups or according to the main tasks of given network hosts. Workstations will produce different usage patterns than server systems in the computer network. Comparing and correlating all workstations belonging to a specific department, therefore, helps to spot suspicious nodes, which have different behavioral patterns.

To address this research gap, we provide an integrated overview and detail system using a novel visualization technique, called *ClockMap*, which uses the approach of circular treemaps as layout algorithm for a large number of temporal glyphs representing

<sup>9</sup> I had the idea of the ClockMap method, which integrates the circular glyph within the seldom used circular treemap layout. The writing, implementation, and programming was done by myself and successfully published at EuroVis [82]. Florian Mansmann and Johannes Fuchs did the proofreading and gave advice. Johannes Fuchs and Christopher Kintzel originally contributed to the glyph design, which was used in matrix layouts in their previous work [140]. Additionally, we presented the approach together with a method to visualize firewall policies by Florian Mansmann [168] on the German IT-Security Congress of the BSI [86].

▼ **Table 3.6** — Related work with methods for host and server monitoring. Overview of related work with respect to data source and visualization type.

Method	Use Case										Data Source										Visualization					Year		
	Internal/External Monitoring	Port Activity Monitoring	Host/Server Monitoring	Packet Traces	Network Flows	IDS/IPS Alerts	Firewall Logs	Vulnerability Scans	Meta Data	System Metrics / Status Reports	DNS Logs	BGP Messages	Server Logs	File System Changes	Audit Trails	Webserver Logs	Database Logs	HoneyPot Logs	Spam / Phishing Mails	Malware Files	Behavior Logs	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display		Dense Pixel Display	Stacked Display
Tudumi [237]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2002
Erbacher et al. [69]	-	-	✓	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	2002
Erbacher [68]	-	-	✓	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	2003
NVisionIP [149]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2004
Portall [77]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	-	2005
Mansman et al. [163]	-	-	✓	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2008
McLachlan et al. [172]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	2008
Pearlman and Rheingans [189]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2008
Phan et al. [191]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2008
Frei and Rennhard [94]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2008
Berthier et al. [21]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2010
Best et al. [24]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2010
ORCA [19]	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	2011
Kintzel et al. [140]	-	-	✓	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	✓	✓	-	2011
Erbacher [71]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2012
Fischer et al. [85]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2012
NV [115]	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	2012
Change-Link [157]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2012
StreamSqueeze [169]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	2012
RainMon [211]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2012
Song et al. [221]	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2012
VAFLÉ [98]	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2013
Hao et al. [112]	-	-	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2013
Hao et al. [113]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2013
ELVIS [121]	-	-	✓	-	-	✓	-	-	-	-	-	✓	-	-	✓	✓	-	-	-	-	-	✓	-	✓	-	✓	-	2013
Change-link 2.0 [156]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	✓	-	-	2013
CORGI [122]	-	-	✓	-	-	✓	✓	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓	-	2014
Visual Filter [224]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2014
Walton et al. [263]	-	-	✓	-	-	-	-	✓	✓	-	-	-	-	✓	-	-	-	-	✓	-	-	✓	-	✓	-	-	-	2014
Kotenko and Novikova [144]	-	-	✓	-	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	✓	-	-	2014
Pixel Carpet [151]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓	-	2014
Zhang et al. [281]	-	-	✓	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2015
Chen et al. [42]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2015
Wang et al. [264]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2015
Ocelot [12]	-	-	✓	-	✓	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	✓	2015



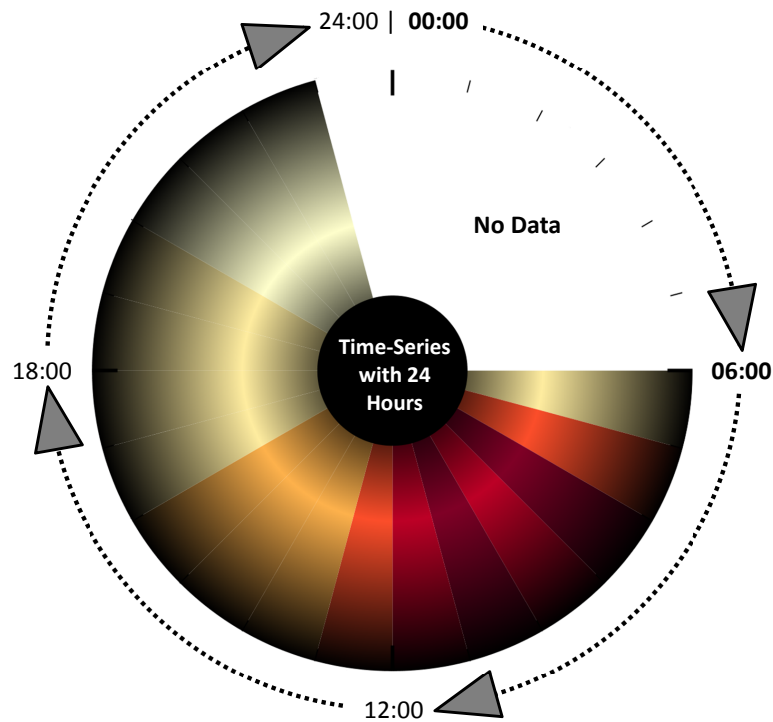
data values of time-series. In particular, we apply this idea to a clock-based glyph inspired by the work of Kintzel et al. [140], which we call *clockeye*. The advantage of this circular design is, that we can smoothly switch between different levels of the hierarchy and either show aggregated overview data for a subnet or show all individual time-series as glyphs.

## Related Work

In the last decade, treemaps [217] became one of the most popular techniques for visualizing hierarchical data. While there are many different treemap types, rectangular treemaps are used most often. Variants of such rectangular treemaps usually represent several data dimensions using area and color of the different rectangles within their actual hierarchy. Much research was conducted in the area of layout algorithms, but also in visual improvements of the different treemap variants. Cushion treemaps [256], for example, use intuitive shading to provide better insights in the hierarchical structure. Since it is often important to compare different treemaps from different points in time, stability is an important criterion of the algorithms. The layout algorithms can be modified to consider such constraints. For example, Mansmann et al. [165] use treemaps to visualize data traffic and use geographic location to optimize the layout. This helps to compare different datasets of different points in time. Other adjustments of treemaps focus on the integration of temporal information within a single treemap to handle hierarchical *time-variant* data. Chin et al. [44] use animation in treemaps to be useful for dynamic data. Other improvements integrate glyphs or small charts to represent additional time-series information for a particular leaf node [208]. Besides of the aforementioned rectangular treemaps other types have been developed like voronoi treemaps [16] and circular treemaps [268]. However, for good reasons the circular treemap has not been frequently used. Circular treemaps waste space, because they do not “*fill the available space completely*” [268], which also means that they “*fill the available space to a varying degree*” [268] and thus introduce imprecision in the aerial representation of the upper levels. In contrast to space-filling techniques, glyph visualizations are suitable representations [266] for many different purposes. Especially to visualize a large amount of multi-dimensional data points or time-series, glyphs are thus widely used. In the essence, our approach is a combination of circular nested treemaps (e.g., *Pebble Maps* [268]) and a clock-like glyph for time-series data (cf. *ClockView* [140]).

### 3.3.1 ClockMap – Visualization Technique for Host Monitoring

The *ClockMap* visualization technique is based on the combination of temporal glyphs, called *clockeyes*, and a circular treemap layout. The basic idea of *clockeyes* is to make use of the metaphor of a classic clock [140]. A circle is subdivided into sectors, each sector representing a time span of one hour. When 24 slices are used, we have a 24-hour clock as seen in Figure 3.21. In this example, there was no data from 00:00 to 06:00 o’clock and from 23:00 to 24:00, which results in a noticeable empty area in the representation. This can be very helpful to find specific patterns without data or zero data values. At one point between 06:00 and 07:00, the time-series seems to start, having high peaks between 08:00 to 09:00 and 10:00 to 11:00. Afterwards a clear downward trend until 23:00 can be observed, while no traffic is visible in the last hour of the day between 23:00 and 24:00.

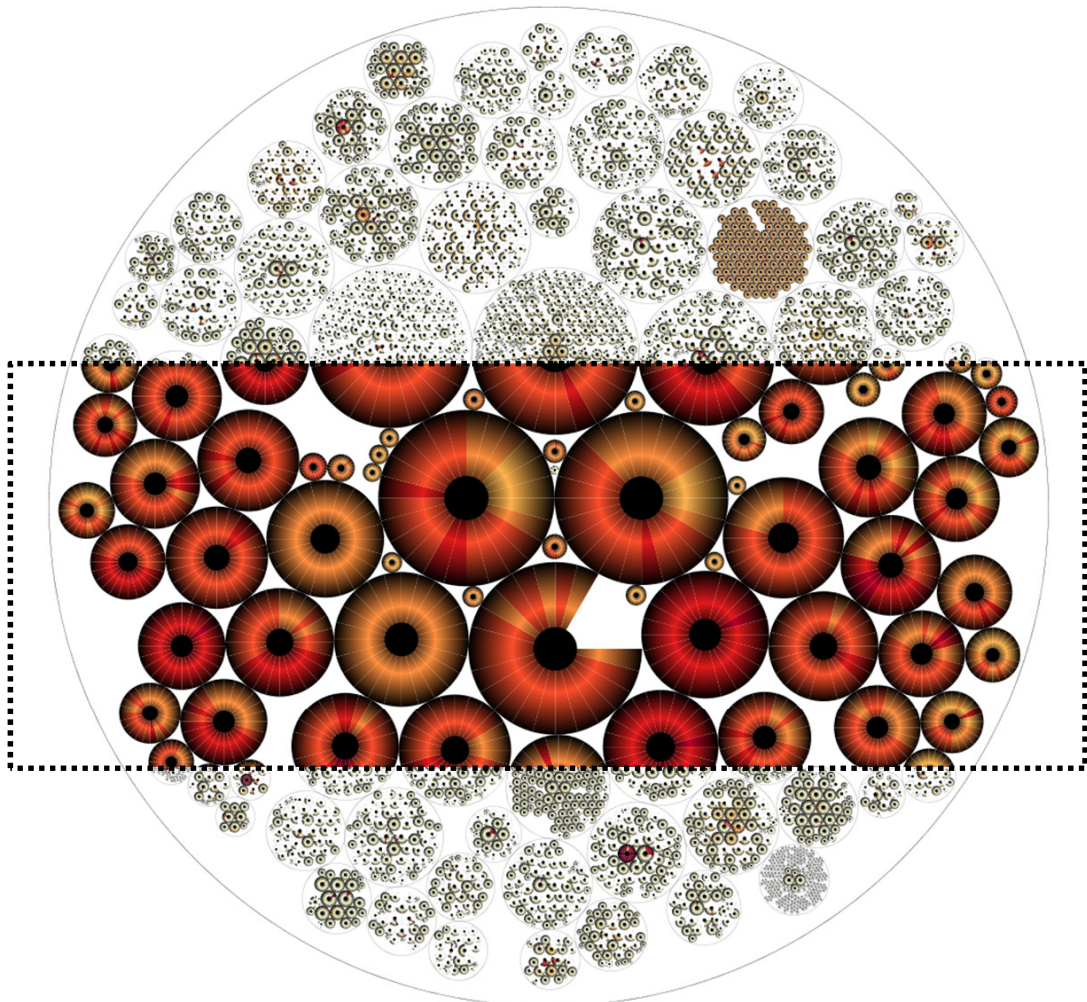


▲ **Figure 3.21** — **Conceptual design of a clock glyph.** Visual representation of a single clock glyph, also named *clockeye*, showing a time-series of 24 hours. Each one hour sector is colored by its data value. Circular shading is applied to emphasize the borders of the glyph. Reprinted from [82]. © 2012 The Eurographics Association.

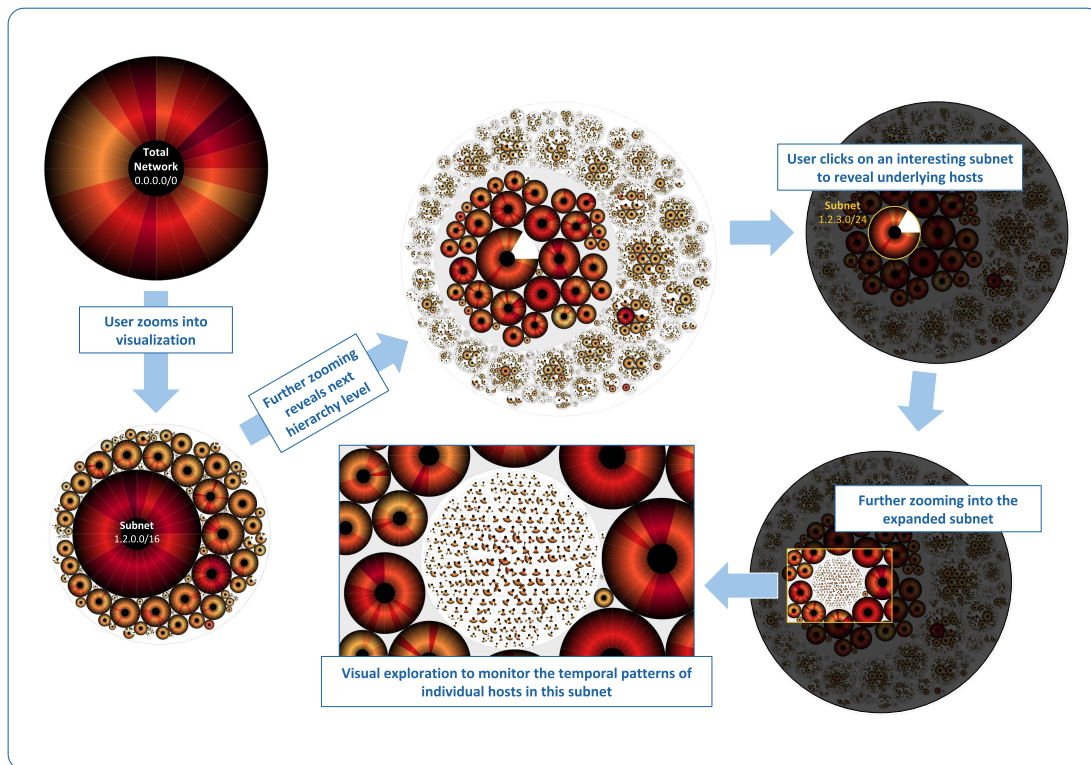
When many *clockeyes* are plotted to a dense area, it is important that they can be separated from each other intuitively, without the need to have an additional border in between. Circular shapes are very suitable for this purpose, because they are perceived as separate items pre-attentively. However, if many have the same color values, this task can become difficult in dense areas. To visually improve the perception of the compactness and further emphasize the borders, we applied circular shading, which seems to be an improvement according to our experiments. This generally led to darker colors, therefore, we decided to use an intense yellow to red color mapping from *ColorBrewer* [31] to counterbalance this effect. The inner black circle can be used for additional meta labels or to indicate highlighting with color.

As discussed there are visualization techniques dealing with hierarchical data and others, e.g., glyphs, displaying temporal or multi-dimensional information. Especially in computer networks the combination helps to understand temporal dependencies in different substructures of the network. With *ClockMap* we use circular treemaps in combination with *clockeyes*. The circular treemap itself is often less powerful than rectangular layouts, however, in the combination with *clockeyes* it seems to be a promising use case. To make further use of the implicit characteristics of the layout algorithm, we implemented *ClockMap* on top of a zoomable user interface, which enables infinite zooming and panning possibilities. Each hierarchy can show the aggregated values for all underlying children to provide the user with a high-level overview as seen in Figure 3.22. While zooming into the aggregated areas more details and eventually each host represented as small *clockeye* becomes visible. Through this semantic zooming,

the scalability of the overall approach is improved, because less visual objects need to be drawn to the canvas when zooming out. Even with thousands of leaf nodes the visualization can be explored interactively. During exploration of real datasets it became obvious that in some cases very prominent nodes need to be removed or moved to another group. To facilitate this, we integrate edit operations to add hierarchies, remove nodes or place them freely into other circles or outside the main circle. After each modification the weights are changed accordingly to automatically recalculate the layout. To search for specific attributes of the nodes, a search field is integrated to *ClockMap*. The black inner circles of matching nodes are highlighted to guide the user to relevant nodes.



▲ **Figure 3.22** — Expanded overview of a whole circular treemap in *ClockMap*. A circular treemap is used to lay out hundreds of *clockeyes* into groups based on their hierarchy. The rectangle illustrates, how the visualization will look like, when the user zooms out. Reprinted from [82]. © 2012 The Eurographics Association.

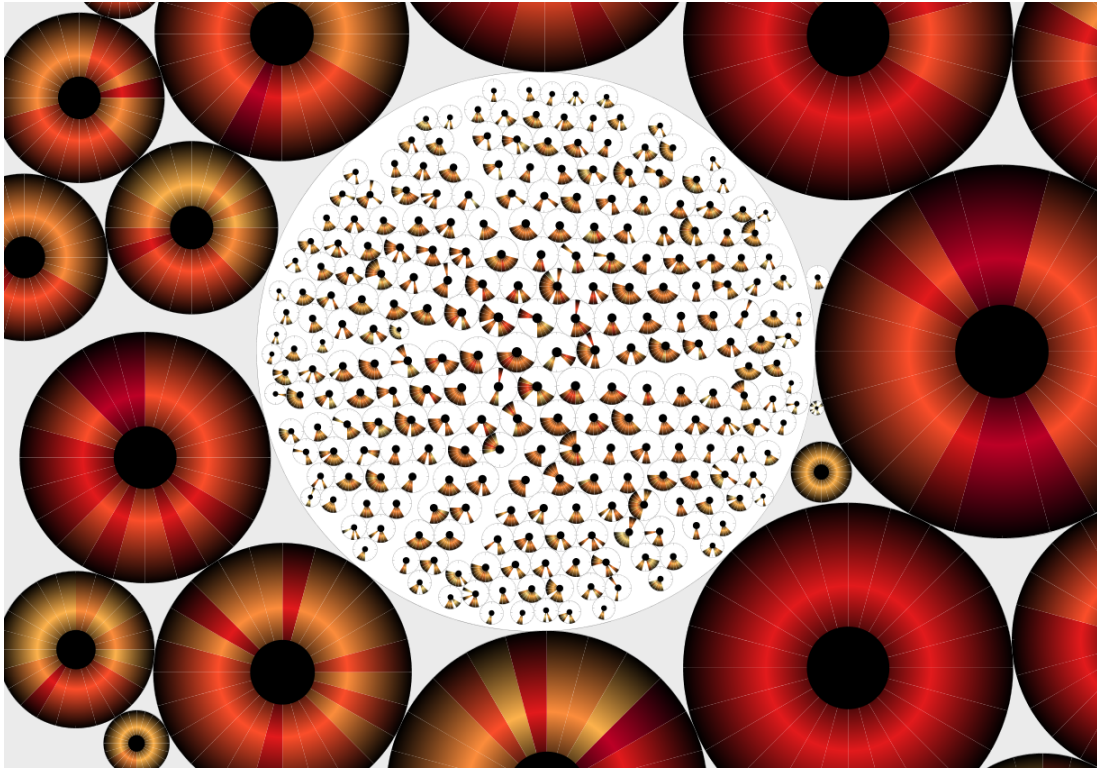


▲ **Figure 3.23** — **Interaction workflow with *ClockMap*.** The interactive workflow in *ClockMap* uses semantic zoom to enhance monitoring and investigation of hosts in a computer network.

### Interactive Exploration using ClockMap

Network operators of large networks often use network flow data for host and server monitoring. Such datasets do not contain payload information, but do contain communication flows between hosts. We used a real dataset of 24-hours with about 200 million (anonymized) NetFlow records collected at the core routers. The data is stored to a database and visually explored with *ClockMap*. The visual analysis does only focus on the records describing the outgoing traffic of all 6,048 hosts belonging to the monitored /16 IPv4 address block, which were active on that particular day. Figure 3.22 shows an example of such a visualization.

The analyst starts with the total network, which is visualized as one single *clockeye* representing the whole computer network (0.0.0.0/0). This workflow is represented in Figure 3.23. The user further zooms in and is interested in the most dominant subnet (1.2.0.0/16). Further zooming reveals next hierarchy levels as seen in the third step of Figure 3.23. Many subnets become visible and one subnet stands out, because of its overall “pacman”-like shape revealing a strange time-series pattern. There was no traffic at all during night hours. This looks suspicious to the analyst, so he further zooms in and clicks on this interesting subnet to reveal the underlying hosts. Further zooming into this expanded subnet reveals more details as also seen in Figure 3.24. This form of details on demand is implemented using semantic zooming. After a user-defined zooming threshold, the time-series for all underlying hosts become visible instead of the previously shown aggregated subnets. Such a pattern could be a network outage



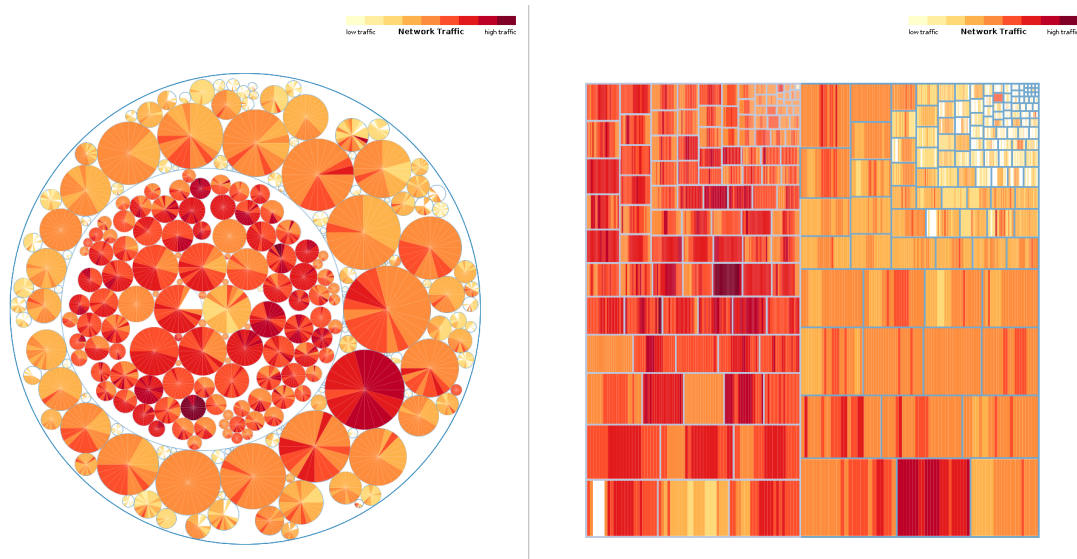
▲ **Figure 3.24** — **Hosts and servers within an interesting subnet.** Underlying hosts of a very prominent subnet having no night time traffic identified using the interactive workflow represented in Figure 3.23.

or indicate a broken switch in the building where the physical machines are located. However, in this case the pattern is legitimate, because it is known as wireless network subnet, which is generally not in use during night time.

### Discussion and Limitations

While the previous section showed the general applicability and usefulness of *ClockMap* to explore large datasets to enhance host and server monitoring, the efficiency and effectiveness needs to be evaluated and compared to other techniques. In the following, we will discuss the advantages and limitations of our approach.

The layout of glyphs is often determined by coordinate systems or matrix layouts. Kintzel et al. [140] use a matrix representation to position IP addresses in a meaningful way. Compared to such matrix layouts, *ClockMap* has several advantages. Matrix representations cannot convey the hierarchy in an intuitive way. The circular treemap layout instead makes the hierarchy obvious, because it is visualized through nested circles. Another advantage is, that the aspect ratio does not change in *ClockMap*. We use circles, which can be further explored through interactive exploration with techniques like zooming and panning. The integration of semantic zooming helps to smoothly switch between general overviews and detailed time-series analysis. Both approaches are overlap-free, while the free arrangement in *ClockMap* results in a tighter packing of the glyphs and thus makes the approach slightly more scalable. In addition, the tight packing better supports the user to visually compare the shapes and color



▲ **Figure 3.25** — **Comparing rectangular versus circular treemap.** Side-by-side comparison of a circular treemap with clock glyphs and a rectangular treemap with embedded striped thumbnail glyphs. The temporal locations are better conveyed in the circular representation, because of the stable aspect ratios compared to the rectangular example.

distributions of neighboring hosts in one branch of the displayed tree. Consequently, outliers with a different behavior in the group can be spotted pre-attentively.

The used *clockeye* glyph also has the advantage to use a common real-world metaphor. Everyone knows how to read a clock, which helps the user to identify particular hour values within the time-series. However, it is even harder to visualize hundreds of different time-series simultaneously. *Clockeye* glyphs are very compact and general trends or patterns can be distinguished even on a very small scale. This helps to provide a scalable way to represent hundreds of time-series, and even more, when grouped within a hierarchy.

Further studies from Diehl et al. [62] suggest to generally use Cartesian coordinate system instead of such a radial display “*unless there are clear reasons to favor a radial one*” [62]. However, the authors also show some beneficial effects of such radial visualizations: “*Memorizing single cells is easier in radial coordinate systems while memorizing three cells is easier in Cartesian coordinate systems*” [62]. Additionally, they state: “*When depicting as many sectors as rings in a radial visualization, sector positions are easier to memorize than ring positions*” [62]. When focusing only on sectors, their study showed, that the study participants got 76% correct answers, compared to about only 70% in the Cartesian alternative [62]. These observations are actual in favor of our *clockeye* glyph, because we are interested in the identification of single time intervals, and use colored sectors only and do not use the mapping of rings at all.

There are also drawbacks of our visualization technique, which are implicit by design. Circular treemaps are indeed not space-filling. This means that, at least compared to rectangular treemaps, space is wasted as seen in Figure 3.25. However, compared to a matrix representation, this is not necessarily the case, because nodes are packed tightly together while still conveying the hierarchy information. The ordering within a group of the circular layout is also challenging and non-intuitive. This drawback

can be overcome to a certain degree by interaction and tooltips. While comparison of shape and color distribution in circular layouts is effective, the comparison of the area of the circles is not. Additionally, the higher level circles only approximately match the aggregated size of their descendants. Consequently, the visualization is probably less precise with respect to these attributes. *Clockeyes* are using color to represent the data values, which makes it hard to precisely compare the values, which would be better in length-encoded glyphs. The basic design idea of *clockeyes* uses a clock metaphor. Obviously, this metaphor cannot be applied any more, if an arbitrary time-series length is used. This means, that a *clockeye* glyph is best suited for 12 or 24-hour time-series. Other lengths of time-series will be less intuitive, but are still possible from a technical point of view.

Figure 3.25 shows a circular treemap with clock glyphs compared to a squarified treemap with embedded striped thumbnail glyphs as introduced in Section 3.1.1. The same dataset is shown in both representations so direct comparison is possible. The drawback of not being fully space-filling in the circular treemap reveals the advantage of implicitly helping to distinguish the various nesting levels. Another advantage is the stable aspect ratio of circle. The changing aspect ratios for the individual rectangles in the squarified treemap, as seen in the right subfigure in Figure 3.25, makes it hard to compare the embedded time-series with each other. While this effect can be minimized algorithmically through further optimizations of the layout algorithms, as proposed by Schreck et al. [208], the problem cannot be resolved completely.

### 3.3.2 Evaluation of Alternative Glyph Designs

In this and the following sections, we present four individual evaluations directly related to *ClockMap*. This section evaluates the clock glyph and compares it with alternative glyph designs for temporal data. Section 3.3.3 summarizes the evaluation approach conducted by Alshaiikh et al. [8] addressing the overall aesthetics of *ClockMap*. Section 3.3.4 integrates and applies *ClockMap* to a big data challenge, in which we use *ClockMap* within a larger system to successfully address VAST Challenge 2012, which focused on large-scale big data analysis. Eventually, we revisit VAST Challenge 2013 in Section 3.3.5 and validate findings and events with *ClockMap*, which were missed with previous techniques as presented in Section 3.1.

The next part builds on the following publication [95]<sup>10</sup>:

J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pages 3237–3246, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi:10.1145/2470654.2466443 [95].

In the previous sections, we made use of mainly three different visual representations to visualize time-series data and used them in a small multiple setting: (i) striped





<sup>10</sup>The responsibilities for this joint publication were divided as follows: Johannes Fuchs and Petra Isenberg designed the user study. Johannes Fuchs and I conducted the experiment. Johannes Fuchs was also responsible for analyzing the results and writing the paper. Petra Isenberg, Florian Mansmann and Enrico Bertini gave advice and did the proofreading.

thumbnail glyphs (STR), (ii) line glyphs (LIN), and (ii) clock glyphs (CLO). The striped thumbnail glyphs are shown in Figure 3.4 and 3.25, line charts are used in various contexts, for example Figure 3.17 and 3.25, while the clock glyph method can also be seen in Figure 3.25.

To compare them in a controlled way, we conduct a formal user study to evaluate the performance of these temporal glyphs in a small multiple setting. Additionally, we also include star glyphs (STA) in the experiment, because the temporal aspect is mapped to angle as it is in clock glyphs. However star glyphs use length instead of color saturation for data value encoding. Table 3.7 shows an overview of the compared glyphs with the different mappings used for temporal and data value encoding. We show combinations of the encodings for quantitative data ranked according to the study results by Cleveland and McGill [48] and highlight some general data density issues.

The results of our study were general enough to derive various design considerations for glyphs representing time-series data. In the following, we will primarily focus only on presenting the results to evaluate the glyph usage with respect to specific temporal locations, which refers to Task 2 in our experiment [95]. All other details, methodology, and extensive discussion of various additional findings can be found in our joint publication [95].

▼ **Table 3.7 — Overview of compared glyphs within the user study.** Partial overview of the design space for temporal glyphs included in the experiment. We show combinations of the encodings for quantitative data (cf. Cleveland and McGill [48]) ranked according to their study results<sup>11</sup>.

Glyph	Temporal Enc.	Data Value Encoding	Data Density Issues
 <b>Line (LIN)</b>	Position CS	Position CS (1)/Direction (3)	May become very dense
 <b>Star (STA)</b>	Angle	Length (3)	Small angular differences are hard to distinguish
 <b>Stripe (STR)</b>	Position CS	Color Saturation (6)	Color blending for small areas
 <b>Clock (CLO)</b>	Angle	Color Saturation (6)	Color blending

## Experiment Results

Within our study we focused on three individual tasks: (1) Peak Detection, (2) Temporal Location, and (3) Trend Detection. In the experiment design we defined the general setup and derived various hypotheses based on two exploratory pilot studies. For each trial, the same type of glyph, but with different data, was drawn on the screen in a small multiple grid layout ( $8 \times 6$ ) showing 48 glyphs in total. Each glyph covered

<sup>11</sup>Ranking based on Cleveland and McGill [48]: 1) Position CS, 2) Position NAS, 3) Length/Direction/Angle, 4) Area, 5) Volume/Curvature, 6) Shading/Color Saturation. Position CS = position along a common scale. Position NAS = position along non-aligned scale.



a fixed screen estate of  $96 \times 96$  pixels. To test the scalability of the different glyph designs, two data densities were tested. The small dataset contained 24 data values. This reflects the standard mapping in *ClockMap* in which each data value maps to the aggregated value for one hour. The large dataset contained 96 data values resulting in a 15 minute resolution. In Task 2 the participants were asked to select the glyph with the highest value at a predefined time-point amongst all shown small multiples. This time-point was textually shown to the participant in advance (e.g. 3am). Therefore, the task involved two mental steps: First the participants had to identify the location of a time-point by making positional (LIN, STR) or angular judgements (STA, CLO). Afterwards, the participants had to compare the peaks. The task consisted of four training repetitions and four real trials for both densities. After the initial training trials we asked participants to detect a different temporal location for the peak value. Therefore, the first real trial was discarded due to the mental recalibration necessary by the participants. In the following, we report the results of our user study for Task 2.

### Accuracy

*“There was a significant effect of glyph on error for both the low density ( $\chi^2(3, N = 32) = 17, p < .001$ ) and the high density condition ( $\chi^2(3, N = 32) = 7.81, p = .05$ ). In the low density condition pair-wise comparisons showed that errors in judgement were significantly worse for LIN (33.3%) compared to CLO (100%) and STA (100%) (both  $p < 0.01$ ) and STR (75%) compared to CLO (100%) and STA (100%) (both  $p < 0.001$ ). In the high density condition STA (58.3%) significantly outperformed LIN (15.5%) and STR (10%) (both  $p < 0.05$ ). With an increasing data density, STA (from 100% to 58.3%), CLO (from 100% to 54.2%) and STR (from 75% to 10%) significantly lost accuracy with  $p < .05$  in each case.*

### Efficiency

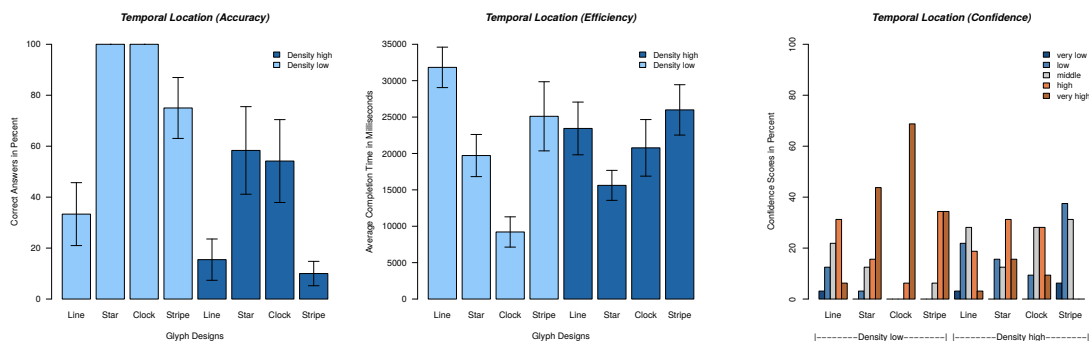
*For the completion time there was only an overall effect of glyph on time in the low density ( $F_{3,21} = 9.1, p < .001$ ) condition. Post-hoc comparisons showed that CLO (9.2 sec.) significantly outperformed LIN (31.8 sec.) ( $p < .01$ ). There was another significant effect of glyph across densities ( $F_{3,21} = 5.45, p < .01$ ). From low to high densities CLO (from 9.2 sec. to 20.8 sec.) deteriorated significantly ( $p < .05$ ).*

### Confidence

*There was an overall effect of glyph on confidence for both the low density ( $\chi^2(3, N = 32) = 13.78, p < .01$ ) and the high density ( $\chi^2(3, N = 32) = 12.12, p < .01$ ) condition. For the low density condition the results showed a clear picture for the confidence of the participants. The users were significantly more confident when using CLO (73.8%,  $p < .05$ ), and had least confidence with LIN (50%,  $p < .05$ ). For the high density condition the subjects were nearly equally confident using CLO (52.5%) or STA (54.4%), whereas LIN (44.4%,  $p < 0.05$ ) and STR (35%,  $p < 0.001$ ) are ranked worst. From low to high densities STA (from 65.6% to 54.4%,  $p < .05$ ), CLO (from 73.8% to 52.5%,  $p < .001$ ) and STR (from 65.6% to 35%,  $p < .001$ ) worsened.” [95]*

In the experiment design, we defined the following hypothesis: “When detecting temporal positions, STA & CLO (angular enc.) outperform LIN & STR (position enc.)” [95]. Our assumption was, that the usage of the familiar clock metaphor is beneficial, so we expected that circular glyphs allow the perception of specific points in time to be more accurate. We assumed that this effect is stronger for CLO than STA as the clock shape is more clearly retained. The results, as shown in Figure 3.26 partially support this hypothesis.

As seen in the first chart concerning accuracy in Figure 3.26, the star and clock glyph outperform the line and stripe glyphs when the dataset was low with respect to accuracy. Star and clock lead to 100% of correct answers. When data density was high (dark blue) we observed the same trend, even though only the star glyph showed significant differences with respect to stripe and line glyph. The good performance of the star glyph can be explained with the combination of the encodings. The length encoding for the data values makes it possible to easily spot the highest value even with lots of datapoints. With the color encodings, participants had problems spotting the peak value. We saw almost no significant differences between the designs for efficiency. The average completion time is reported in the second chart of Figure 3.26. However, the clock glyph was significantly better than the line glyph with low data density. Furthermore, as visually represented in the third chart of Figure 3.26, participants were significantly more confident and made significantly less mistakes with the polar designs. The participants also reported to like the clock metaphor. Some suggested, however, to visualize only 12 hours at a time for a more intuitive encoding [95].



▲ **Figure 3.26** — Evaluation for peak comparison on temporal locations. The summary of results for our conducted user study. It reveals differences between various glyph designs with respect to accuracy, efficiency, and confidence. The clock glyph had relatively high accuracy, with good efficiency (low completion time, especially for low densities), and achieved high confidence scores for temporal peak locations tasks.

### 3.3.3 Evaluation of ClockMap’s Design Principles

The evaluation of design decisions, especially with respect to aesthetics, is often biased by the author’s subjective opinion, which often relates to aesthetics found in nature. Additionally, studies show that aesthetics influence efficiency and effectiveness of analysis tasks [40]. In 2013, Alshaikh et al. [8] proposed a “novel evaluation approach for security visualization based on Christopher Alexander’s fifteen properties of living structures” [8].

Alexander [6] identified these fifteen properties of order in the context of art and nature: strong centers, level of scale, boundaries, alternating repetition, positive space, good shape, local symmetries, deep interlock and ambiguity, contrast, gradients, roughness, echoes, the void separateness, simplicity with inner calm, and non-separateness [6]. These “*properties are derived from various visual patterns that appear in nature. Each property represents the guidelines for good design*” [8]. Alshaikh et al. [8] “*believe that using these fundamental properties have the potential for building a more robust evaluation. Each property offers essential qualities that enable better analytical reasoning*” [8]. Interestingly, Alshaikh et al. [8] used our *ClockMap* approach as an example and evaluated and compared *ClockMap* with another visualization system for cyber security. In the following, we present the summary of their findings when they evaluated *ClockMap* according to the properties of order by Alexander [6]. They could actually find nine out of Alexander’s fifteen properties (AFP) in *ClockMap*, while none of the properties could be identified in the second approach under investigation:

*“We believe that ClockMap’s ability to represent the network activity data lays in its compactness and clarity due to its effective use of size, shape, and color. But underlining ClockMap’s effectiveness is its compliance with Alexander’s fifteen properties.*

*ClockMap used STRONG CENTRES represented in the circles as areas show subnet activity. Using LEVELS OF SCALE, ClockMap shows the variation between subnets as circles take different sizes. THICK BOUNDARIES is used in the relatively thick edge formed around each circle. At the centre of each circle lies a black dot, or a circle, that represents THE VOID. ClockMap makes use of GRADIENTS within each circle moving gradually from red to yellow. The crescent shape around the clustered circles in the middle forms a GOOD SHAPE. The edges of ClockMap forms a THICK BOUNDARY using ECHOES and STRONG CENTERS. In the middle of the ClockMap, two large circles apply LOCAL SYMMETRIES around the vertical axis, and another two circles form symmetry around the horizontal axis. But in both cases, the symmetry is not perfect. Notice that in both cases one of the circles is smaller than the other, and in one case the circle on the right is incomplete. So, there is an element of ROUGHNESS. The crescent on the edge of the circle is not shaped by the clustering circles in the middle alone, but by the POSITIVE SPACE surrounding it as well. The ClockMap is an example of how to use the AFP to produce effective visualizations. (...) The properties within the visualization work together to form a living structure that responds to the context of the system, while maintaining beauty and clarity.” [8]<sup>12</sup>*

This positive feedback of other researchers analyzing our *ClockMap* method from the perspective of design, highlights the visibility of our approach in the field of cyber security, but also validates various design decisions made during sketching, designing, and implementing *ClockMap* from a more formal design-oriented perspective.

<sup>12</sup>This paragraph is taken from the publication by Alshaikh et al. [8]. They are researchers with background in computer and network security. I’m not affiliated with them. The authors evaluate our approach completely on their own based on my publication about *ClockMap* [82], which was previously published to share the technique with other researchers in a timely manner. Furthermore, we made a simplified version available online, so they obviously could use the technique on their own, without the need to re-implement the method.

### 3.3.4 Evaluation using VAST Challenge 2012<sup>13</sup>

The sections coming next build mostly on the following publications [83, 90]<sup>14</sup>:

F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: A Visual Situational Awareness Tool for Large-Scale Computer Networks (VAST Challenge 2012). In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 257–258, 2012. doi:10.1109/VAST.2012.6400528 [83].

F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: Visual Analytics for Big Data in Large-Scale Computer Networks. *Information Visualization*, 14(1): 51–61, 2015. ISSN 1473-8716, 1473-8724. doi:10.1177/1473871613488572 [90].

In this section, we focus on *BANKSAFE*, which integrates and applies *ClockMap* to a big data challenge, in which we use *ClockMap* to successfully address VAST Challenge 2012, which focus on large-scale big data analysis. *BANKSAFE*, therefore, provides a proof-of-concept with respect to scalability, of using *ClockMap* to actively solve a big data analysis challenge. *BANKSAFE* was the award winner for an “*Outstanding Comprehensive Submission*” amongst a total of “*40 submissions with participants from 12 different countries*” [52], while the datasets were downloaded by almost 1,100 people [52]. The VAST Challenge 2012 explicitly focused on scalability relating to visual analytics for big data:

*“The goal of VAST Challenge 2012 was to provide a set of realistic computer network scenarios while pushing the boundaries of big data. The setting of the Challenge is BankWorld, a planet much like Earth, but with a very different geography. For this Challenge, the geography is one large land mass containing several different nation-states. The most important organization on BankWorld is the Bank of Money (BOM). BOM has many offices of various sizes across BankWorld. Each of these offices has many computers active throughout the day. In total, the organization operates about 895,000 machines. Contestants were asked to focus on two general problems using a visual analytics approach. First, how do you achieve cyber situation awareness across the entire enterprise with such a large number of systems? Second, when something does go awry, can you identify it and the steps needed to resolve the problem?.”* [52]

In the given scenario, the so-called “Bank of Money” operates in “BankWorld” and has collected a dataset for two different challenges. This dataset comprises of 4.1 GB of IDS alerts and firewall logs, and 7.5 GB of health and status checks data of a host monitoring solution. These health checks are generated every 15 minutes by over 895,000

<sup>13</sup> <http://vacommunity.org/VAST+Challenge+2012>

<sup>14</sup> *BANKSAFE* was mostly implemented by myself, while Johannes Fuchs and Florian Mansmann contributed to various individual parts, like the activity-policy matrix. Together, we submitted our solution to the VAST Challenge 2012 leading to a short paper [83], in which all authors were involved for writing, discussing, and proof-reading. The success of *BANKSAFE* led to our joint journal article [90], which highlights the lessons-learned. The writing was mostly done by myself. Florian Mansmann focused on proof-reading, while Johannes Fuchs and Daniel Keim contributed with discussions and comments.

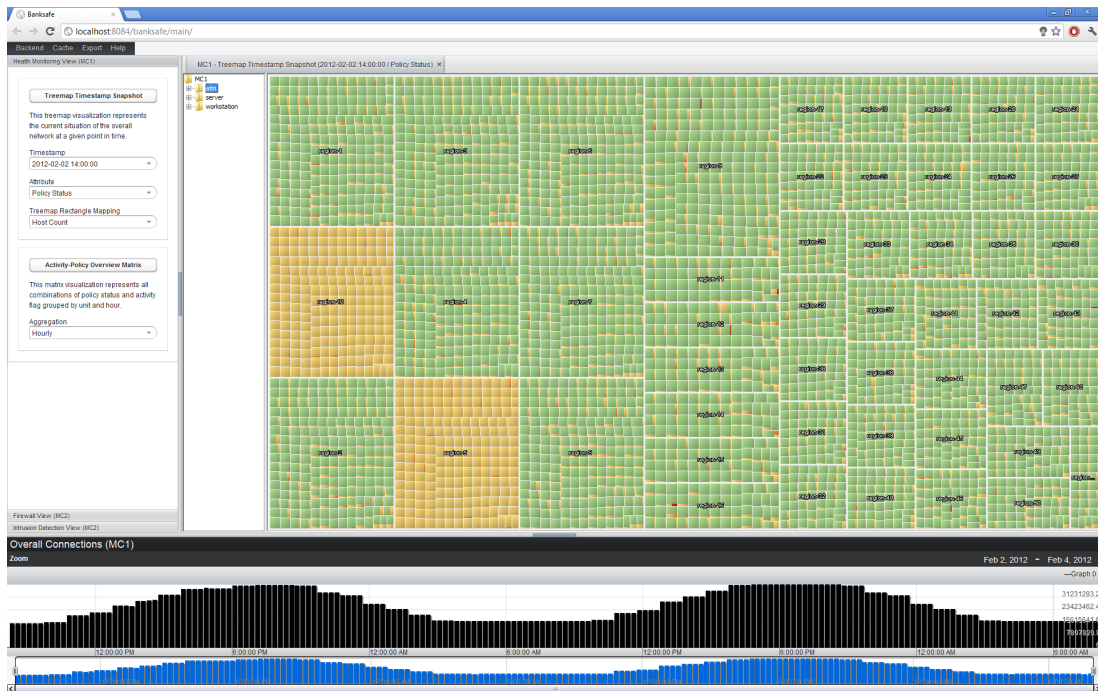
machines [52] and were the focus of the first challenge (MC1). The second challenge (MC2) provides the security-related IDS alerts and firewall logs of one particular regional office. The main task of MC1 was to create suitable visualizations to provide situational awareness to understand the network health and identify problems of this global large-scale computer network. MC2 required the identification of unusual and suspicious events to propose countermeasures for a large-scale regional office.

### *BANKSAFE – A Scalable Visual Analytics System using ClockMap*

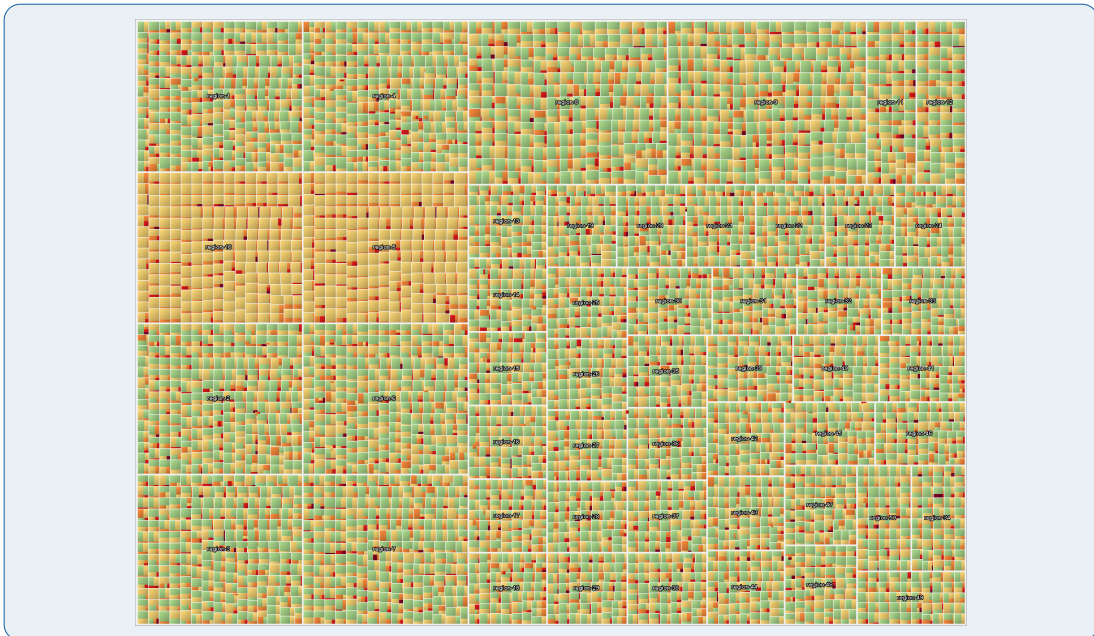
To actively participate and compete in the challenges, we integrate our existing techniques as discussed in the previous sections, and implemented *BANKSAFE* as seen in Figure 3.27. To achieve scalability for large datasets our system makes use of the cloud-based database service Google BigQuery [103]. Monitoring and security data are imported into this backend. The remote data storage is accessed via an API. The main application is developed as Java web application hosted by Apache Tomcat. To further improve performance and to reduce costs, all queries are routed through a high-performance caching system. Additionally, *BANKSAFE* provides a web-based graphical user interface using the Vaadin Java Web Framework [254]. The different visualization modules are implemented using Java Applets, HTML5, and *D3.js* [29]. Besides standard bar charts to represent the number of active hosts or events, *BANKSAFE* includes several visualizations to support the analyst in getting an overview, finding trends, and identifying suspicious events. We developed different visualizations specifically for monitoring data, but also integrated *ClockMap* as novel visualization approach.



▲ **Figure 3.27** — *BANKSAFE* in a control room scenario. The usage the system in a control room setting helps to analyze big data in large-scale computer networks to achieve situational awareness. Reprinted from [90]. © 2013 The authors.



▲ **Figure 3.28 — Point-in-time network health overview.** This treemap visualization provides an overview for the current state of the whole computer network. Reprinted from [90]. © 2013 The authors.

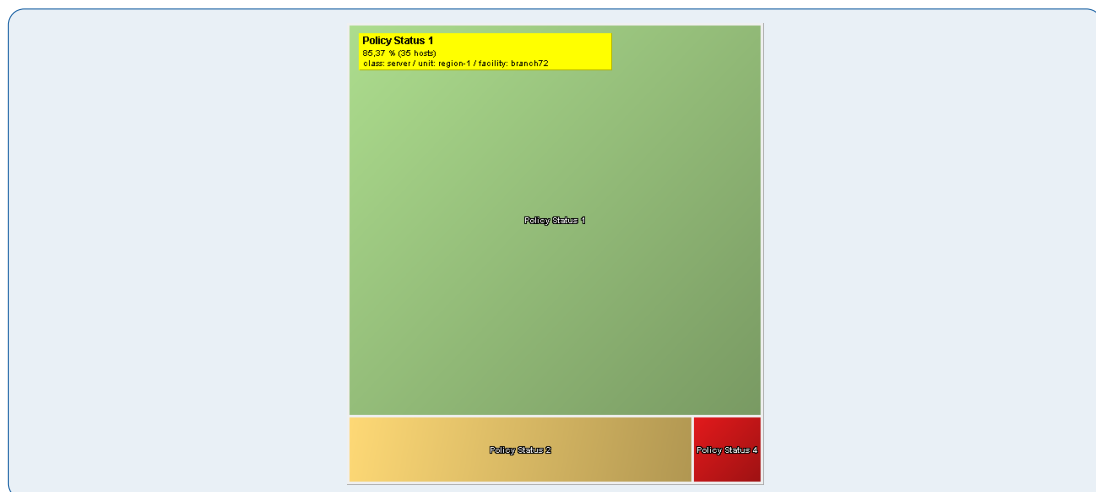


▲ **Figure 3.29 — Network health overview during an infection.** This case represents a wide-spread computer infection leading to a high percentage of critical policy levels. Reprinted from [90]. © 2013 The authors.

### Usage Scenario and Analysis Workflow for VAST Challenge 2012

When an user starts the web application, three views are loaded on the left sidebar (cf. Figure 3.28). Each view is related to the main tasks for a different type of data. The network health view focuses on monitoring data, the firewall perspective is linked to network traffic, and the intrusion detection view makes use of event-based alerts generated by intrusion detection systems (IDS). Depending on the selected view, there are different configuration settings available. The user can select a data source and time interval to load a suitable visualization module, which is added as tab to the main display. Additionally, a time chart is shown at the bottom of the web application, representing the number of hosts or events over time depending on the selected view. With the help of our web application, it is possible to get a visual overview of heterogeneous datasets to enhance situational awareness. In the following, we describe the various visualization components used to solve the VAST Challenge 2012:

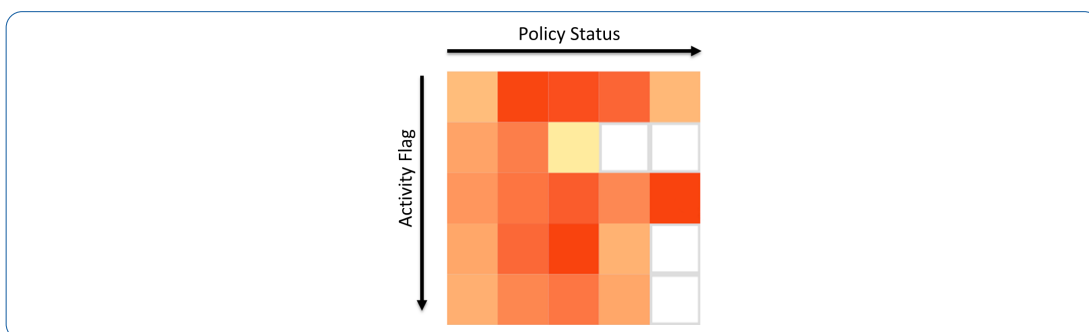
- **Network Health Overview** – The network analyst needs to have a point-in-time network health overview to be aware of the current situation of the overall network. In the challenge’s dataset, each computer has a policy status ranging from 1 to 5, while 5 is the most critical one, indicating a possible virus infection. The network is structured in classes, regions and facilities, leading to an organizational hierarchy. Consequently, conveying this hierarchy in the visual representation helps the analyst to detect abnormal behavior. Figure 3.30 for example presents the distribution of policy levels as treemap of all about 50 servers in a single selected facility. About 85 percent of the hosts do have a policy status of 1, represented as the most prominent green rectangle, while just a few have higher policy levels, visualized as smaller yellow and red rectangles. This simple, but space-filling and scalable representations can give the analyst a point-in-time overview. When this is applied to all regions and facilities in the network, patterns and suspicious regions can be visually identified and explored as seen in Figure 3.28.



▲ **Figure 3.30** — Treemap to convey a facility’s policy distribution. A treemap visualization showing the percentage of computers with different policy levels in a single facility and region. *Reprinted from [90]. © 2013 The authors.*

Based on this visualization the analyst can understand the current network situation on 2012-02-02 at 14:00, which was a particular task in the challenge. The overall impression is quite acceptable, because the green color, representing a good system health status of most machines, is the dominating color in most areas. Further filtering could be used to hide low policy levels, and just focus on the infected hosts. However, visualizing the context is important to come to the right conclusions. In this case for example, *region-5* and *region-10* visually stick out, because here the yellow color is dominating, which means, that almost all machines suffer from moderate policy deviations (no green rectangles at all). Showing all regions is helpful for the analyst to judge such findings with respect to the patterns in all other regions, which further supports the conclusion that these regions have very suspicious computer health distributions. At a later point in time, the analyst wants to get another network overview, which is shown in Figure 3.29. The visual impression is completely different than the previous one: Many yellow, and even red-colored rectangles indicate a wide-spread infection of thousands of computers at 2012-02-04 03:30.

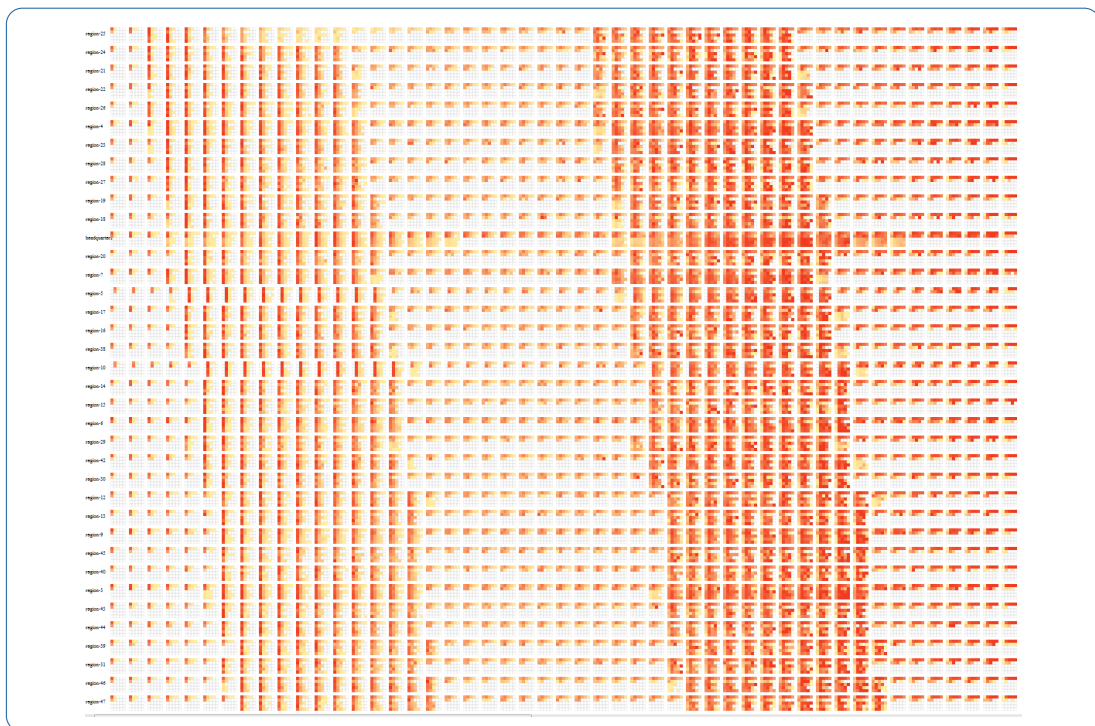
- **Temporal Network Health** – Another task is the identification of possible trends in the monitoring data. To provide a compact, but high-density information display, we implemented the following matrix idea. A single  $5 \times 5$  colored pixel-matrix is depicted in Figure 3.31. Each pixel represents the number of underlying hosts. The matrix shows all possible combinations of policy level and activity scores of a single region for one hour. This means that the yellow pixel at position *row 2 column 3* represents the number of hosts having activity flag 2 and policy status 3. To get a temporal overview, these matrices are arranged in a small multiple display where each row represents a single region and the different columns the different hours. Additionally, the ordering of the rows is done according to a 1D MDS projection of the geographic coordinates of the respective headquarters. Geographically near regions are plotted near to each other. As a result, as seen in Figure 3.32, clear temporal patterns can be identified. With the help of this representation, the general trend of a continuous shift to higher policy levels and more activity became visible. Furthermore, many machines in the East suddenly went offline, which was another unexpected and very suspicious finding.



▲ **Figure 3.31** — **Activity-policy matrix visualization.** Colored rectangles represent the number of hosts having a particular activity flag and policy combination. Reprinted from [90]. © 2013 The authors.



- **Relaxed Timeline for IDS Alerts** – To analyze IDS events, the analyst can use the *Relaxed IDS Timeline*, similar to the one used in previous work [85]. Events are plotted on different timelines. Each timeline contains the events of a particular source IP address. The color is mapped to the event classification attribute, which helps to visually distinguish the event types. Selecting an event presents more information and highlights all other events of this particular type using connecting lines. With the help of this visualization for event data, several hosts producing IRC authorization messages could be identified. It seemed that those machines became suddenly infected and attempted to talk with their bot master over IRC.
- **ClockMap for Firewall Data** – To visualize time-series data of many hosts within their respective subnet or organizational hierarchy, we use the *ClockMap* visualization technique. In the second mini-challenge, the analyst need to solve the task to identify suspicious events and connections. Figure 3.33, for example, shows the traffic of all computers within the network, connecting to an IRC service on port 6667/TCP. This technique was also used to identify forbidden SSH connections, which were initiated by intruders. Interactively exploring this compact glyph representation is, therefore, capable of answering questions like: Which host has suspicious connections to a specific port? Which subnets are affected? What is the connection pattern? How much traffic do they produce?



▲ **Figure 3.32** — Small multiple representation of activity-policy matrices. Each row represents all hourly activity-policy matrices for a given region in a small multiple setting to get an overview about temporal developments of network health.

## Evaluation Results

The visualizations integrated in *BANKSAFE* helped to identify trends, patterns and suspicious events in both datasets of the VAST Challenge 2012. We successfully used *BANKSAFE* to actively participate and address the challenge. In the following, we quote the review process and afterwards present the scores as given by the anonymous reviewers:

*“Including both the visualization community reviewers and the subject matter expert reviewers, a total of 102 reviewers participated, each providing between one and five reviews. (...) Reviewers were asked to rate the analytic process, the visualizations, the interactions, the clarity of explanation, and the relative novelty of the submission. (...) Reviewers provided both ratings and explanatory comments. These comments were as important as the scores in identifying award candidates. Reviewers were also asked to evaluate the plausibility of the answers provided, rather than the accuracy of the solutions. The datasets used this year were realistically complex. Although there were certain known patterns embedded in the data, the committee recognizes the likelihood that additional patterns exist in the data that were not intended and that could reasonably be considered by the participants to be of significance. Consequently, reviewers were provided with a list of the expected patterns that were embedded in the dataset to support the scenario, but they were also instructed to accept other solutions for which the submission provided well-reasoned supporting evidence. The VAST Challenge Review Committee held a one-day meeting to determine awards. Prior to the meeting, all of the committee members examined at least nine of the submissions in detail, with five committee members examining all 40 submissions. During the meeting, the committee reviewed and evaluated the award recommendations from the reviewers, taking the totality of the scores and reviewer comments into account.” [52]*

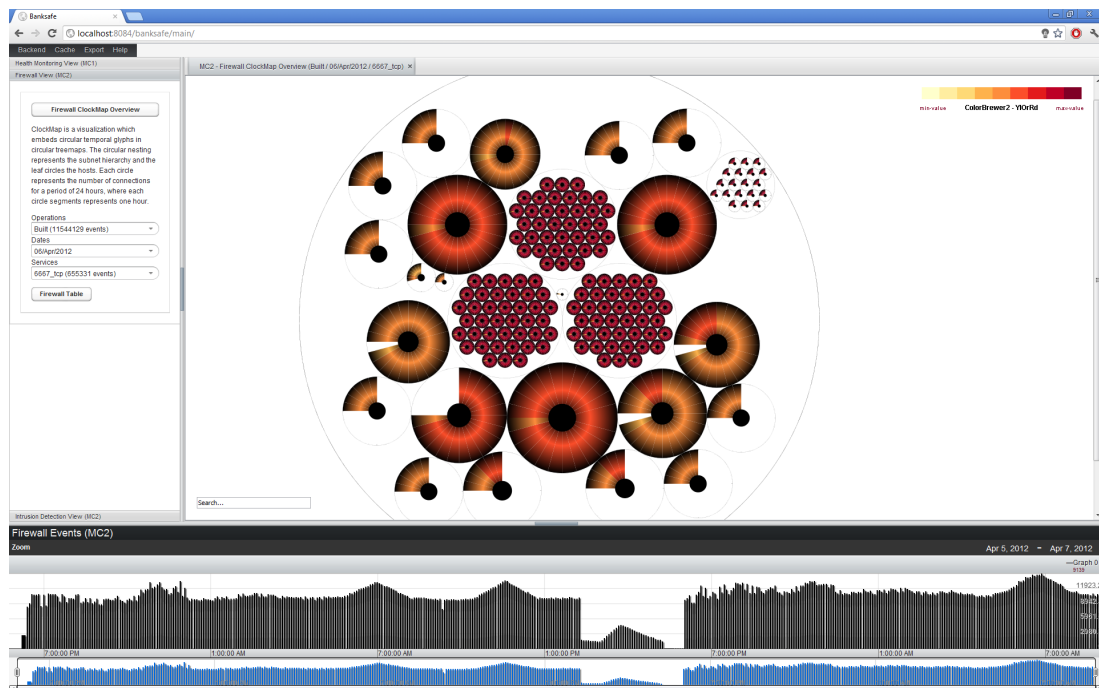
Our submission was judged by seven anonymous reviewers, who provided a summary and detailed written comments about the various fields of interests. Table 3.8 summarizes the general scores as provided by the reviewers (R1-7) for our VAST Challenge 2012 solution with submission ID #118 focusing on MC2 featuring our *ClockMap* approach. We also successfully submitted a solution for MC1, which primarily used the other visualizations (e.g., network health overviews as seen in Figure 3.28 and 3.32), however these techniques are not the primary concern of this section’s evaluation.

As presented in Table 3.8, all reviewers, except Reviewer 4 (R4), gave an overall rating of “good” or “excellent”. Sadly, R4 was the only reviewer not providing any comments or details supporting or explaining the scores. 5 of 7 reviewers also explicitly acknowledged the novelty of the *ClockMap* approach, while R7 commented: “I felt like I was battling the visualizations rather than having them help me” (R7). Probably, more training or a more detailed explanation of the visualization techniques would have helped to address such issues (which was not possible due to the limitations of the overall submission length). However, R7 still acknowledged that almost “all events” were successfully identified. R3 wrote: “The authors correctly identified the illegal introduction of a botnet to the network, leading to high TCP traffic in general and to suspicious SSH traffic over night. For the latter, their time-based clockmap visualization turns out to be very helpful!” (R3). R2 states that, the “*ClockMap* visualization is probably the most informative” (R2).

▼ **Table 3.8** — Summary of reviewer scores for *BANKSAFE*. The table presents the scores given by the anonymous reviewers (R1-7) for our VAST Challenge 2012 solution with submission ID #118 focusing on MC2 featuring our *ClockMap* approach.

	R1	R2	R3	R4	R5	R6	R7
Reviewer Type:	external	external	external	external	external	external	external
Identification of Events:	some events	some events	most events	most events	most events	most events	all events
Identification of Trends:	average	good	average	good	excellent	excellent	excellent
Root Causes:	good	good	excellent	average	good	excellent	average
Clarity of Explanation:	good	good	excellent	good	excellent	excellent	average
Visualizations:	good	good	excellent	average	excellent	excellent	marginal
Interactions:	average	good	good	average	good	excellent	average
Novelty:	good	average	excellent	good	excellent	excellent	marginal
Overall Rating:	good	good	excellent	average	excellent	excellent	good

R3 also stated: “I especially liked how the clockmap helped in detecting the suspicious temporal SSH patterns” (R3). R6 was impressed by the interaction capabilities and stated: “The video showed examples of navigating from overview visualizations to details interactively. It was clear that patterns observed at overview levels could be explored and investigated down to the record level. The interactive response time shown in the video was impressive for a dataset of this size” (R6).



▲ **Figure 3.33** — *ClockMap* visualization in *BANKSAFE*. The different colored circles represent local computers establishing connections to IRC servers. The colored segments of each circle represent the number of connections over time. *Reprinted from [90]. © 2013 The authors.*

### 3.3.5 Evaluation using VAST Challenge 2013<sup>15</sup>

In this section, we revisit VAST Challenge 2013, as introduced in Section 3.1.2. While it was not possible to detect various suspicious (including many subtle) events with the common visualization techniques originally included in *VACS* (Table 3.2), we use the *ClockMap* visualization approach to explore the VAST Challenge 2013 dataset again to see, which of the missed events, can be easily detected when utilizing *ClockMap* instead. The findings are summarized in Table 3.9.

▼ **Table 3.9 — Ground truth evaluation for *ClockMap*.** The ground truth for the VAST Challenge 2013 MC3 consists of 29 official events and various bonus events. This table provides an overview about only those events missed in *VACS* as previously shown in Table 3.2. Events, which are quite obvious in *ClockMap* are marked with ✓, while those which are harder to catch with (✓). The reference value indicates the total percentage of how many submissions in the challenge actually identified the event.

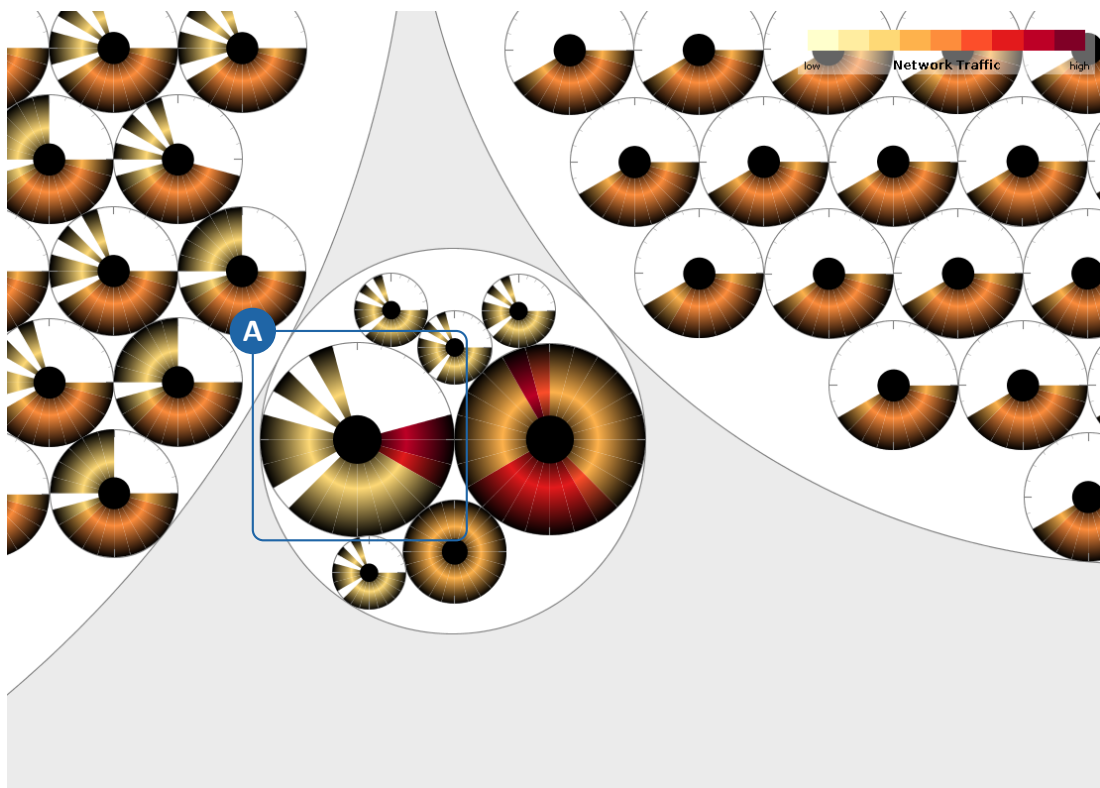
Event ID	Subtlety	Event Type	Data Source	Reference	ClockMap	Figure
(6a)	Subtle	Server Crash	NetFlow/BB	0.0%	(✓)	Fig. 3.34
(6b)	Subtle	Server Return	NetFlow	0.0%	(✓)	Fig. 3.35
(7)	Subtle	Port Scans	NetFlow	0.0%	✓	Fig. 3.38
(9a)	Subtle	Server Crash	NetFlow/BB	36.4%	✓	Fig. 3.36
(9b)	Subtle	Server Return	NetFlow	36.4%	(✓)	Fig. 3.37
(10)	Subtle	Malicious Redirects	NetFlow	0.0%	×	-
(11)	Obvious	Exfiltration	NetFlow	18.2%	✓	Fig. 3.43
(14)	Obvious	Exfiltration	NetFlow	18.2%	✓	Fig. 3.44
(17)	Obvious	Port Scans	NetFlow/IPS	27.3%	✓	Fig. 3.39
(18)	Obvious	Port Scans	NetFlow/IPS	22.7%	✓	Fig. 3.40
(19)	Obvious	Failed DoS	NetFlow/IPS	36.4%	✓	Fig. 3.40
(21)	Obvious	Port Scans	NetFlow/IPS	18.2%	✓	Fig. 3.40
(22)	Subtle	Botnet Infection	NetFlow	9.1%	(✓)	Fig. 3.45
(24)	Obvious	Port Scans	NetFlow/IPS	9.1%	✓	Fig. 3.41
(25)	Obvious	Port Scans	NetFlow/IPS	18.2%	✓	Fig. 3.42
(26)	Obvious	Botnet DoS Attacks	NetFlow/IPS	18.2%	(✓)	Fig. 3.46
(27)	Obvious	Botnet DoS Attacks	NetFlow/IPS	9.1%	(✓)	Fig. 3.47

#### Analyzing the Impact of Denial-of-Service (DoS) Attacks

When monitoring the internal hosts and servers, peaks in network activity can directly be seen in *ClockMap*. Figure 3.34 focuses on the subnet with various web servers on 2013-04-02 based on the feature `firstSeenSrcTotalBytes` of the underlying network flow data. Visualizing them in their context, helps to judge the peaks at specific point in times with respect to other hosts in the same organizational level. The host annotated with (A) in Figure 3.34 reflects the hourly network traffic of 172.30.0.4 (WEB03.BIGMKT3.COM), which is the company’s main server. The analyst can utilize mouseover tooltips to reveal details about a host underneath the mouse pointer. The

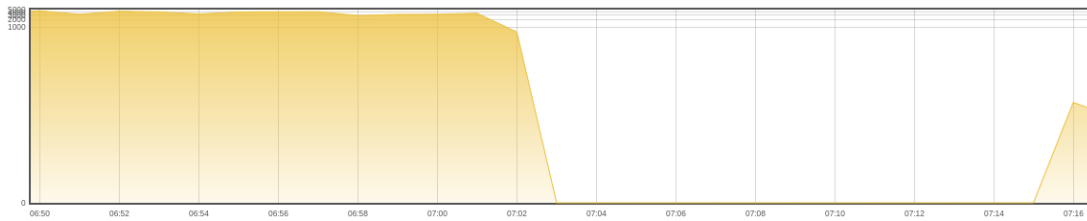
<sup>15</sup> <http://vacommunity.org/VAST+Challenge+2013>

extreme peak between 05:00 and 07:00 refers to a massive denial-of-service (DoS) attack (discussed as Event 5 in Section 3.1.2). Interestingly, network activity is suddenly decreasing starting between 07:00 and 08:00, which is quite unusual for the company’s main webserver, leading to the assumption that this server cannot handle the load any more (Event 6a). However, there is still some traffic visible in the following hours (Event 6b), therefore, the server is probably not completely unavailable or started to resume operations. To confirm the actual health status of the server, a more detailed exploration is needed. The visualization in Figure 3.34 only uses an one hour resolution, however, when using minute intervals for each clock segment, it would be visible that the server returns to operate on 07:16 as seen in the timeline plot shown in Figure 3.35. Therefore, *ClockMap* can somewhat identify that the server returns to operation, but not exactly when, because of the currently used hourly aggregation level.

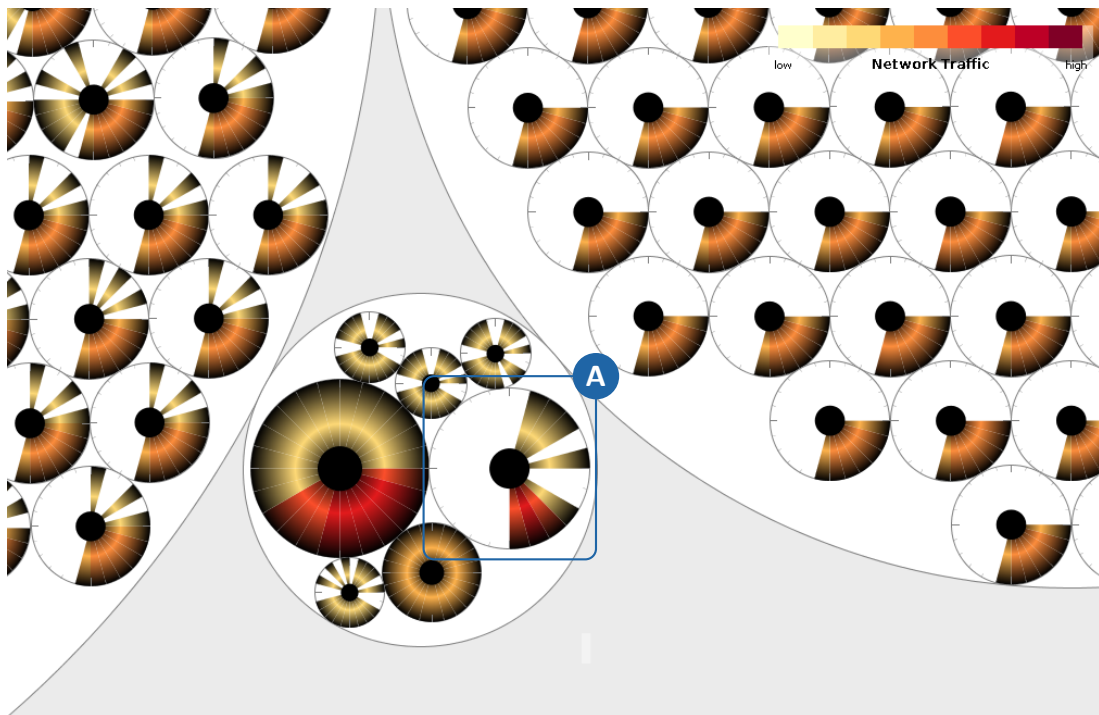


▲ **Figure 3.34** — Degraded network activity after DoS attack. The visualization shows network traffic on 2013-04-02, in which (A) represents 172.30.0.4 (WEB03.BIGMKT3.COM). Extreme peaks between 05:00 and 07:00 are caused by an ongoing DoS. However, the network traffic suddenly decreases between 07:00 and 08:00, and stays on a very low level (light yellow) which are symptoms for major server issues (Event 6a).

On 2013-04-03, it is possible to observe another major server crash between 11:00 and 12:00 (Event 9a) after a high traffic peak caused by another DoS attack. Figure 3.36 shows the obvious pattern of 172.30.0.4 (WEB03.BIGMKT3.COM), in which no traffic at all can be seen after the crash. Actually, the server comes back online not until two days after this event (Event 9b) as shown in Figure 3.37.



▲ **Figure 3.35** — Detailed temporal network activity for main webserver. High amount of network traffic originating from 172.30.0.4 until 07:02. No response until 07:16. Afterwards the web server recovers and resumes operations (Event 6b).

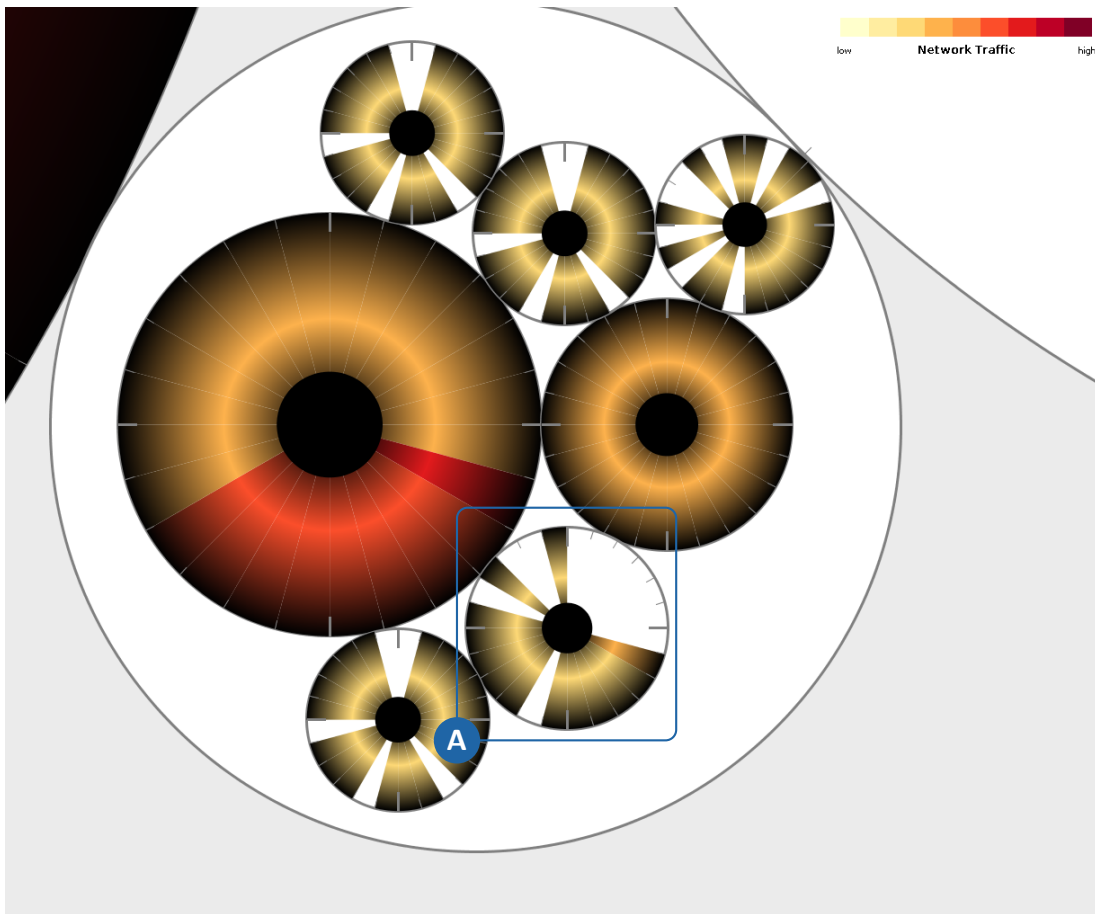


▲ **Figure 3.36** — Visualization for server crash on 2013-04-03. (A) represents the host 172.30.0.4 (WEB03.BIGMKT3.COM) having lots of network activity followed by a long period without network traffic, indicating a major server crash (Event 9a). Other hosts in the subnet obviously do not experience such outages.

Another denial-of-service attack can be seen in Figure 3.40 on 2013-04-11 in the clock segments between 11:00 and 13:00, in which many external hosts (attackers) have a high number of distinct source ports. However, contrary to the previous DoS attack the server withstands the attack and continues to operate normally.

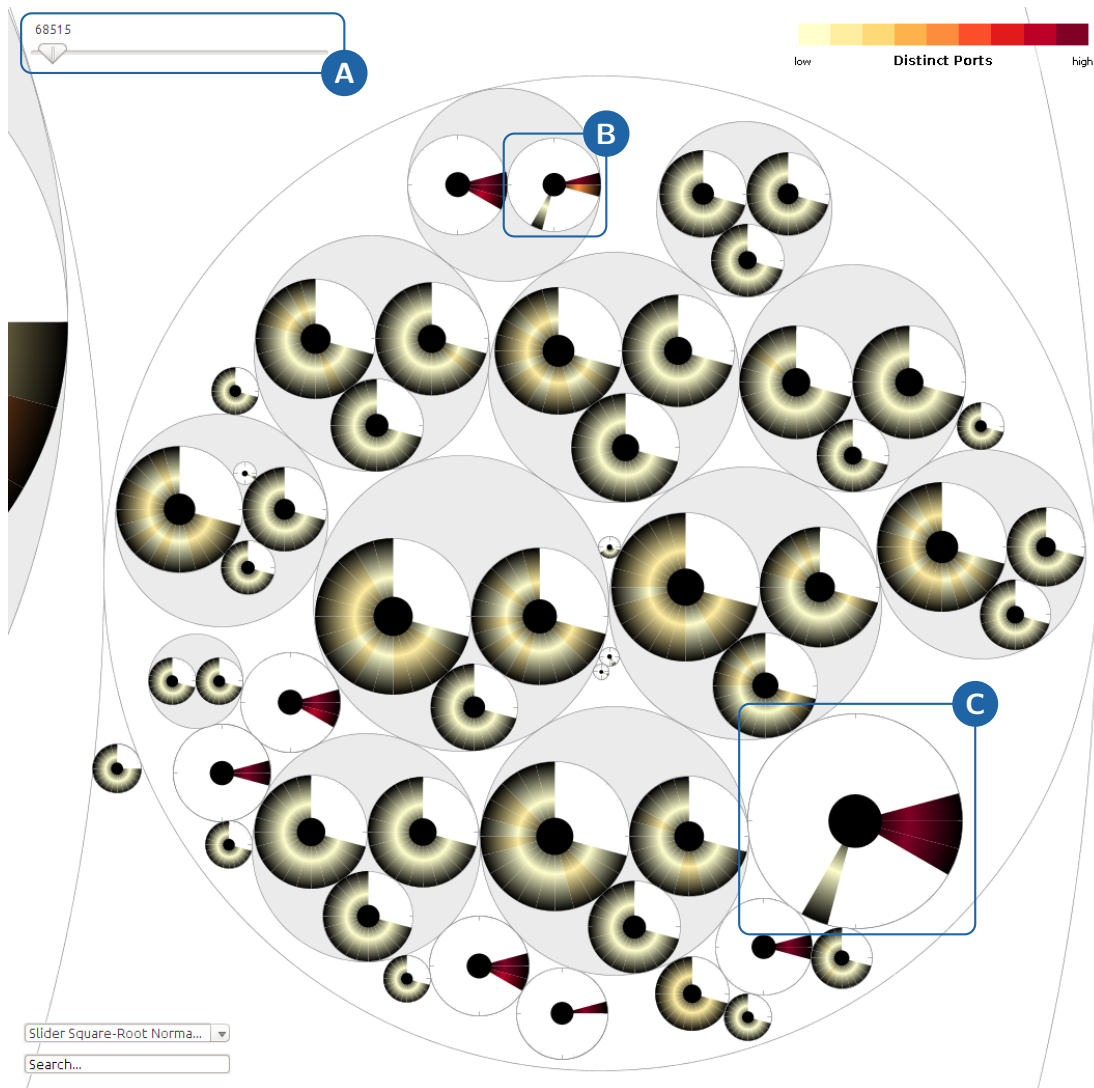
### Detection of Subtle Port Scans

Analyzing the external network hosts on 2013-04-02 reveals an interesting pattern as seen in Figure 3.38. The visualization shows the distinct count for the feature `firstSeenSrcPort`. This feature refers to the number of distinct source port numbers seen in the network flow data within a specific hour of a host. Many opened distinctive



▲ **Figure 3.37** — Visualization of server return on 2013-04-05. (A) represents the host 172.30.0.4 (WEB03.BIGMKT3.COM), which was offline after a server crash and returns back to operation between 07:00 and 08:00. Please note, that compared to Figure 3.36 the host’s location within the visualization has changed, because of the heavily reduced overall network traffic.

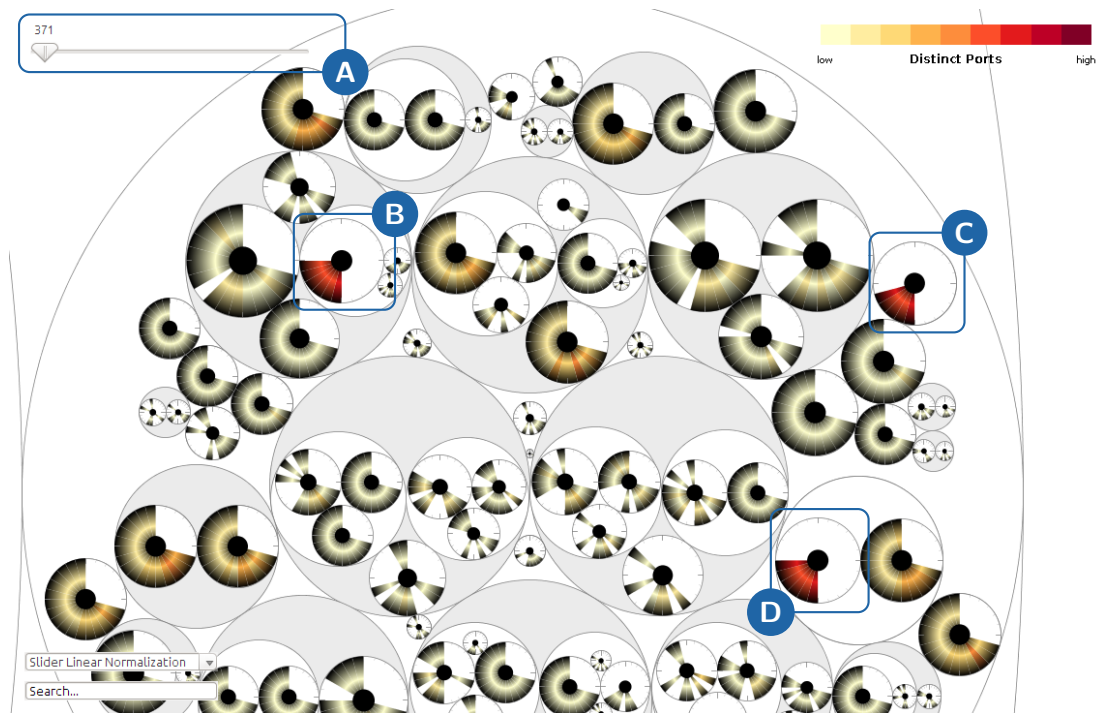
ports from an external host could mean, that many connections are established to internal hosts to check for running services (e.g., using a port scan). Depending on the amount of distinct source ports, this can also relate to connections from a DoS attack. To focus on hosts with high numbers of distinct ports, the analyst can use the normalization slider (A) to interactively change the value at which a segment is visualized using a dark-red color. Switching the normalization technique from linear to square-root further helps emphasize some variations in more detail. Multiple hosts pop up immediately, which have high number of distinct ports between 05:00 and 07:00. However, a more subtle pattern appears for two of those hosts: 10.7.7.10 (B) and 10.6.6.6 (C) which are clearly suspicious have another peak between 13:00 and 14:00 while all other “red-colored” hosts don’t have any other communication beside the major attack in the morning. Obviously, those two hosts reveal a different attack pattern or conduct an additional attack (in this case port scans) between 13:00 and 14:00. This highlights the capabilities of *ClockMap* to show suspicious patterns within the context of other related hosts, to successfully identify such subtle attacks as well. Actually, as can be seen in Table 3.9 none of the challenge submissions (reference value of 0.0%) was capable to identify this event (Event 7).



▲ **Figure 3.38** — **Detection of subtle port scans.** The visualization of distinct ports on 2013-04-02. Using the normalization slider (A), subtle port scans from 10.7.7.10 (B) and 10.6.6.6 (C) between 13:00 and 14:00 can be identified (Event 7), which are not part of the more obvious DoS attack (Event 5) between 05:00 and 07:00.

Focusing on the hourly number of distinct ports for external hosts, reveals more port scans on the following days. After using the normalization slider (A) in Figure 3.39 on 2013-04-10, multiple attackers stand out (Event 17), annotated with (B-D). The prominent clock pattern, helps to assume that this is most likely an orchestrated port scan, because they conduct their port scans in the same time window, with a similar strength. Port scans for longer period of time become obvious on 2013-04-11, in which two hosts stand out in Figure 3.40. While attacker 10.12.15.152 (A) and 10.6.6.7 (B) reveal distinctive patterns, they still share the same pattern of conducting port scans over a longer period of time. The attack of (A) is also verified by the ground truth as Event 18, while the port scan of (B) reflects Event 21. Various other high-volume port scans can be identified on 2013-04-12 (Event 24) and 2013-04-13 (Event 25) as seen in Figure 3.41 and 3.42.





▲ **Figure 3.39** — Multiple attackers conducting orchestrated port scans. Various port scans (Event 17) from multiple attackers on 2013-04-10 are clearly visible. Using the normalization slider (A), the hosts stand out revealing quite distinctive patterns than related hosts in the respective subnets.

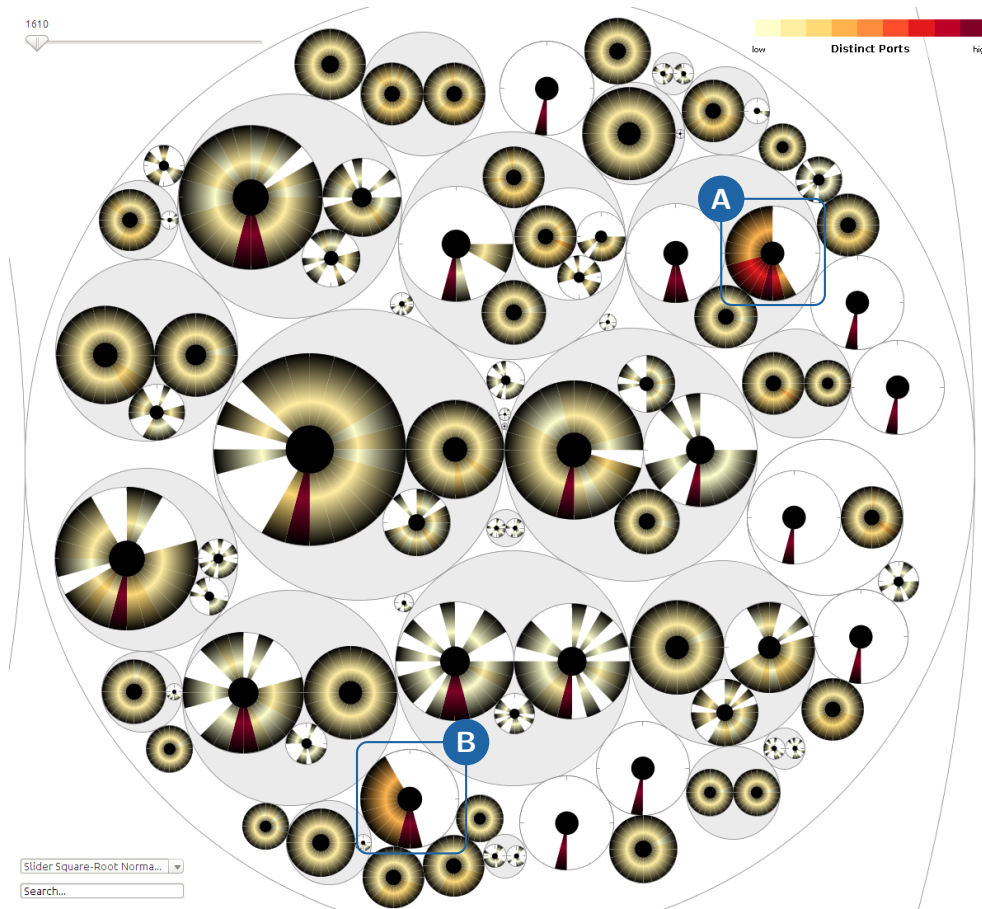
### Detection of Malicious Redirects

Malicious redirects are hard to analyze on a flow-based level, because possible signals are very subtle. Therefore, Event 10 could not be detected easily with *ClockMap*, without actual knowledge about the event. The ground truth reveals, that one of the webservers, “*www.bigmkt2.com is compromised and a malicious redirect is added to a web page*” [210]<sup>16</sup>, so that all “*visitors to www.bigmkt2.com are redirected to a malicious web site, where the visiting computers can also become infected with malware*” [210]. Event 10 could have been detected if someone would use *ClockMap* to show the session durations, but normally there is no good reason to do so. However, the ground truth reveals for Event 10, that because of the malicious redirects, network flows for this host “*exhibit shorter session durations and smaller payloads*” [210].

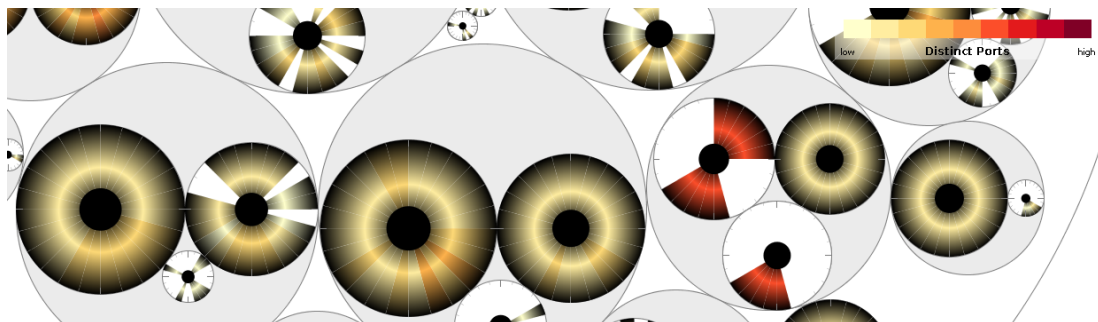
### Identification of Data Exfiltrations

Visualizing the network activity on 2013-04-06 reveals various peaks in network activity. With the help of *ClockMap* the root causes can be identified using interactive zooming and exploring the underlying hosts within the different subnets to reveal possible outliers within their contexts. Expanding subnet 172.10.0.0/24, as seen in Figure 3.43, reveals that most of the traffic between 10:00 and 11:00 originates from a single host, which is the administrator’s workstation (Event 11). Further exploration of the underlying

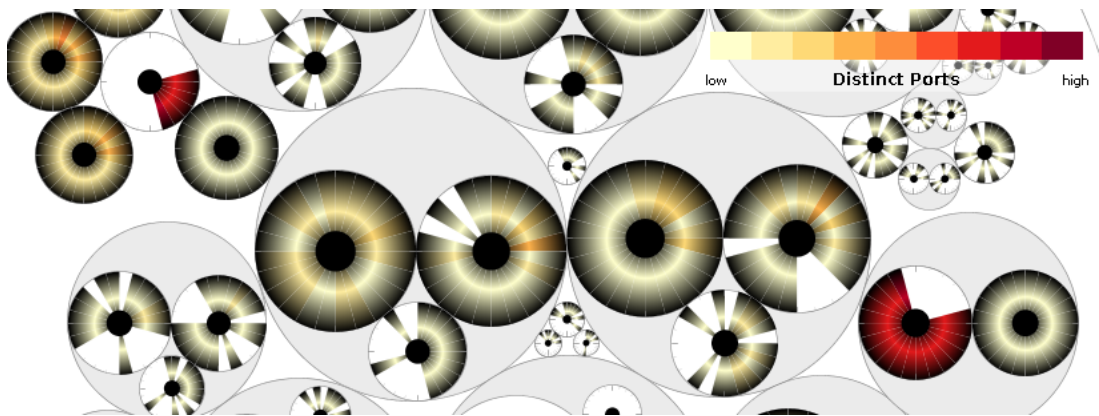
<sup>16</sup>The ground truth can be found in the Visual Analytics Benchmark Repository [210] under Benchmarks / VAST Challenge 2013 / MC3 - Big Marketing / Solution.



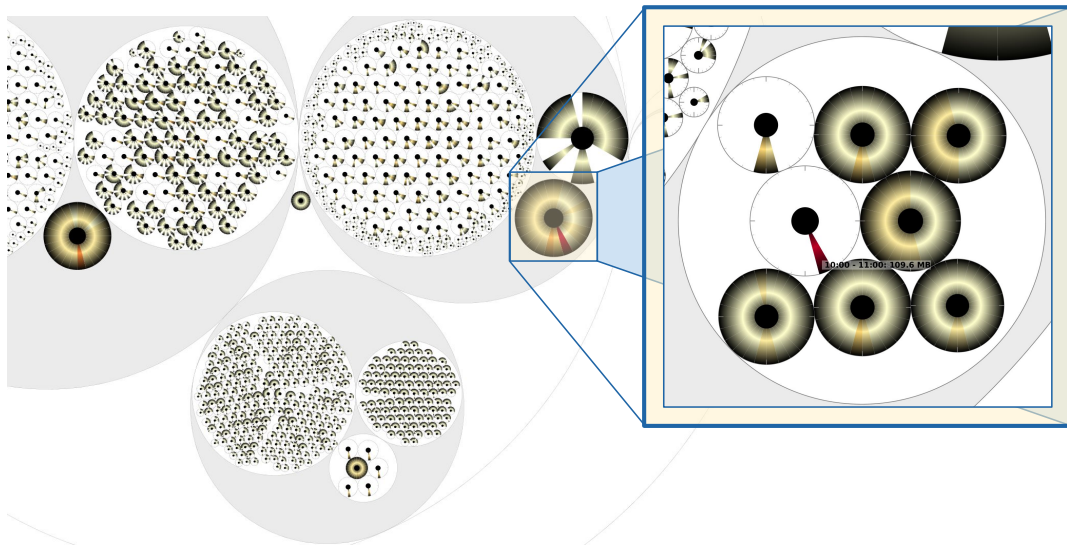
▲ **Figure 3.40** — Identification of port scans over longer periods of time. The high peaks between 11:00 and 13:00 on 2013-04-11 are symptoms of a DoS attack (Event 19). Additionally, various port scans from attacker 10.12.15.152 (A) and 10.6.6.7 (B) over multiple hours relate to Event 18 and 21.



▲ **Figure 3.41** — Port scans of attackers belonging to the same subnet. Port scans from 10.12.15.152 and 10.12.14.15 between 2013-04-12 11:00 and 16:00 (Event 24).

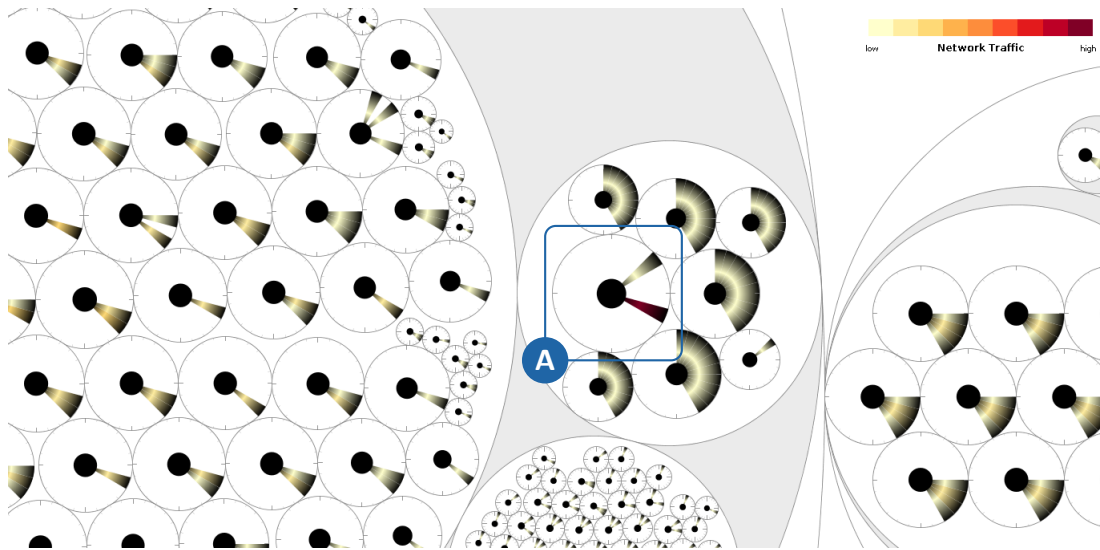


▲ **Figure 3.42** — Port scans of attackers originating from different subnets. Port scans from 10.17.15.10 and 10.12.15.152 stick out with mostly red segments (high number of distinct utilized ports) starting at about 2013-04-13 05:00 (Event 25).

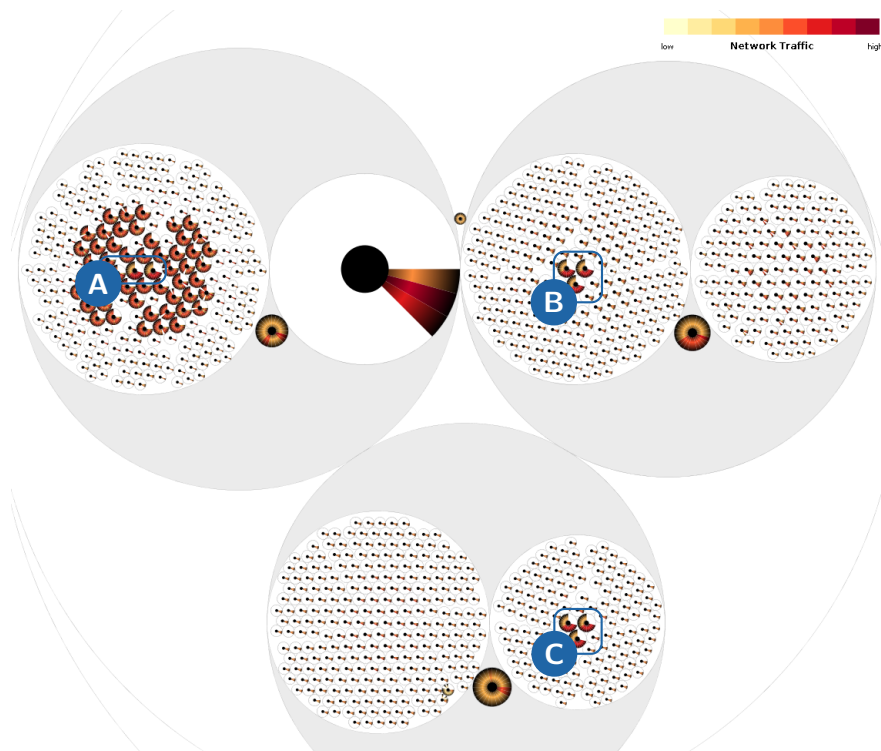


▲ **Figure 3.43** — Root cause identification of high network activity. Exfiltration (Event 11) becomes visible on 2013-04-06 between 10:00 and 11:00. High network traffic in the subnet 172.10.0.0/24 (expanded in the highlighted rectangle) originates from a single host, which is the administrator’s workstation. Further exploration of the underlying network flow records reveals a data exfiltration to 10.7.5.5 of 109.6 MB via file transfer protocol (FTP).

network flow records, reveal a successful data exfiltration to 10.7.5.5 of 109.6 MB via file transfer protocol (FTP). A similar exfiltration can be seen on the day after (2013-04-07) between 07:00 and 08:00 in Figure 3.44 (Event 14), in which an even larger file (about 650 MB) is exfiltrated to 10.7.5.5.



▲ **Figure 3.44** — Identification of large data exfiltrations. Another exfiltration becomes visible on 2013-04-07 between 07:00 and 08:00 using *ClockMap* (Event 14). The high traffic in the subnet 172.10.0.0/24 originates the administrator’s workstation (A). Further exploration of the underlying network flow records reveals another data exfiltration to 10.7.5.5 (about 650 MB).

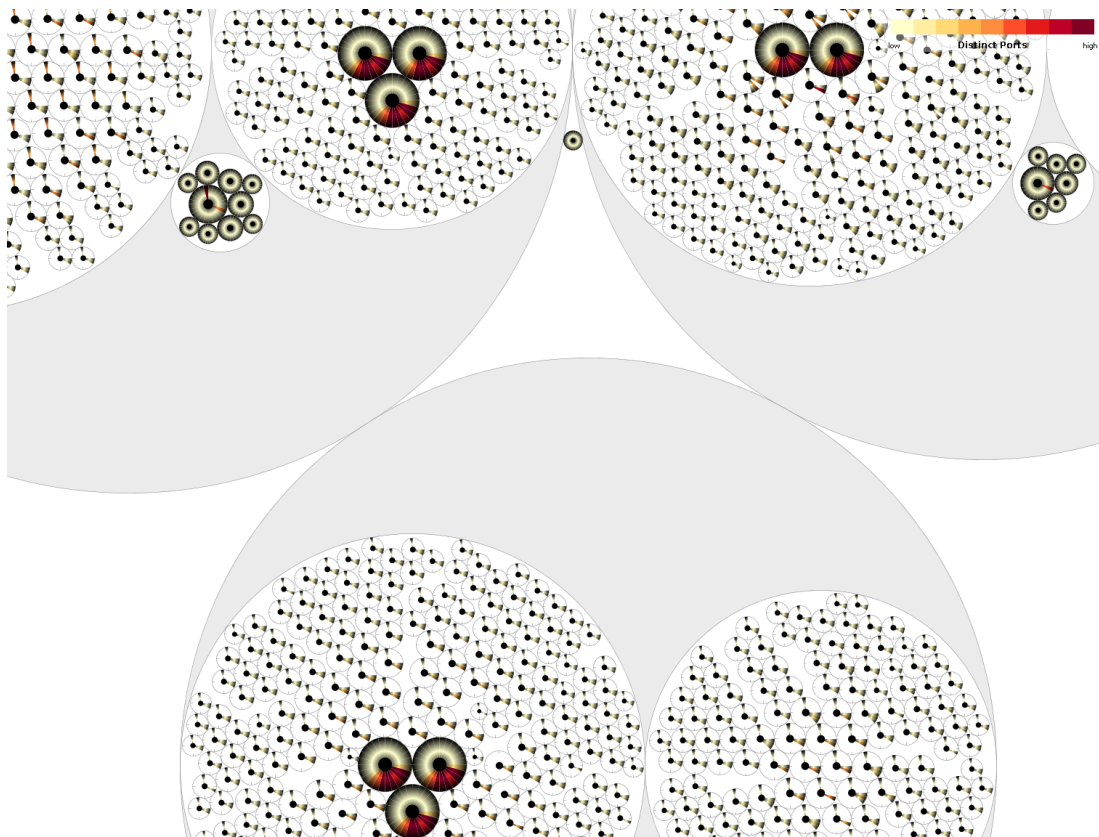


▲ **Figure 3.45** — Identification of outliers in various subnets. Compromised hosts become visible after a successful malware infection (Event 22). In subnet 172.30.1.0/24 two hosts stand out with their pattern (A). In subnet 172.20.1.0/24 three hosts (B) are visible very prominently, and another three hosts (C) are in the focus of 172.10.2.0/24.

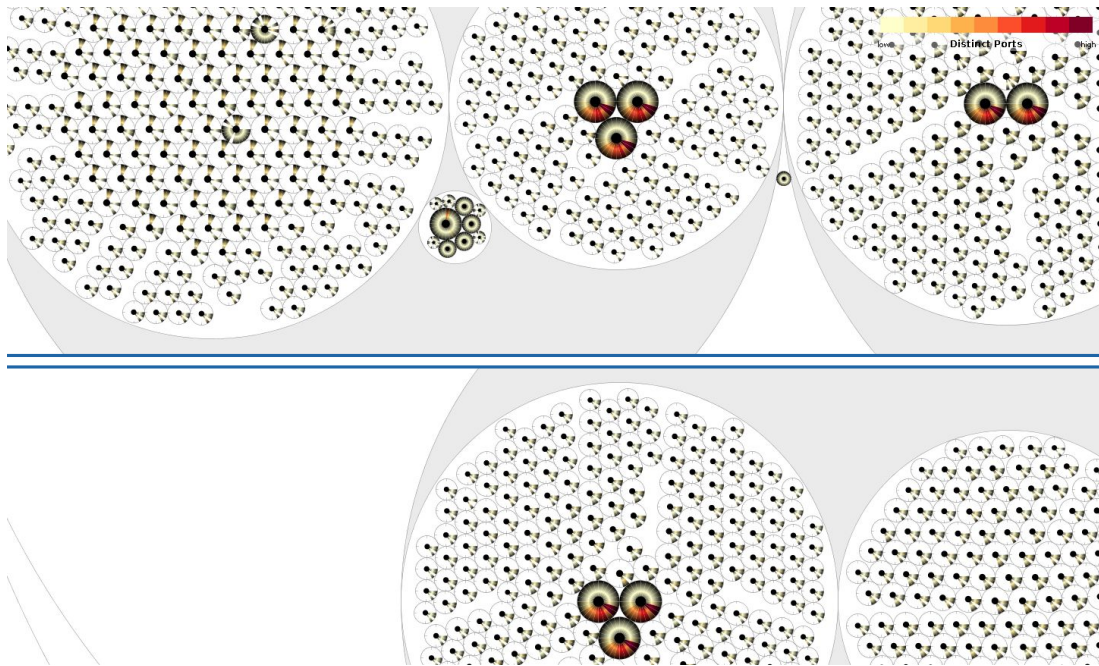
### Identification of Botnet Infection and Compromised Hosts

Taking a look at the network activity of company’s main internal subnets, in which most of the workstations are located, reveals interesting patterns on 2013-04-12. In each of the main subnets, very few hosts behave differently than the rest of the hosts within their contexts. In subnet 172.30.1.0/24 there are many hosts with much network activity, however two still stand out with their pattern (A). In subnet 172.20.1.0/24 three hosts (B) are visible very prominently, and another three hosts (C) are in the focus of 172.10.2.0/24. Something seems to be wired with those hosts. Further exploration of the underlying network flows reveals many SSH connections suddenly appearing starting between 08:00 and 09:00 and continuing throughout the day (Event 23), which relates to the finding presented in Figure 3.14 within Section 3.1.2. Obviously, the hosts became compromised and started to contact their command and control server. While, we could not directly see the very subtle initial infection, *ClockMap* was helpful to still identify the hosts due to the immediate symptoms of the successful infection (Event 22).

In the following days, on 2013-04-13 (Figure 3.46) and 2013-04-14 (Figure 3.47), the infected machines, now part of a botnet, start to become clearly visible with respect to the hourly number of utilized distinct source ports. Further investigations on the aggregated network flow data for the selected hosts, show that these internal hosts target an external webserver, conducting an orchestrated distributed DoS attack, which relates to Event 26 and 27 in the ground truth data.



▲ **Figure 3.46** — Distributed DoS by internal hosts on 2013-04-13. Eight internal hosts start conducting an attack targeting an external webserver (Event 26).



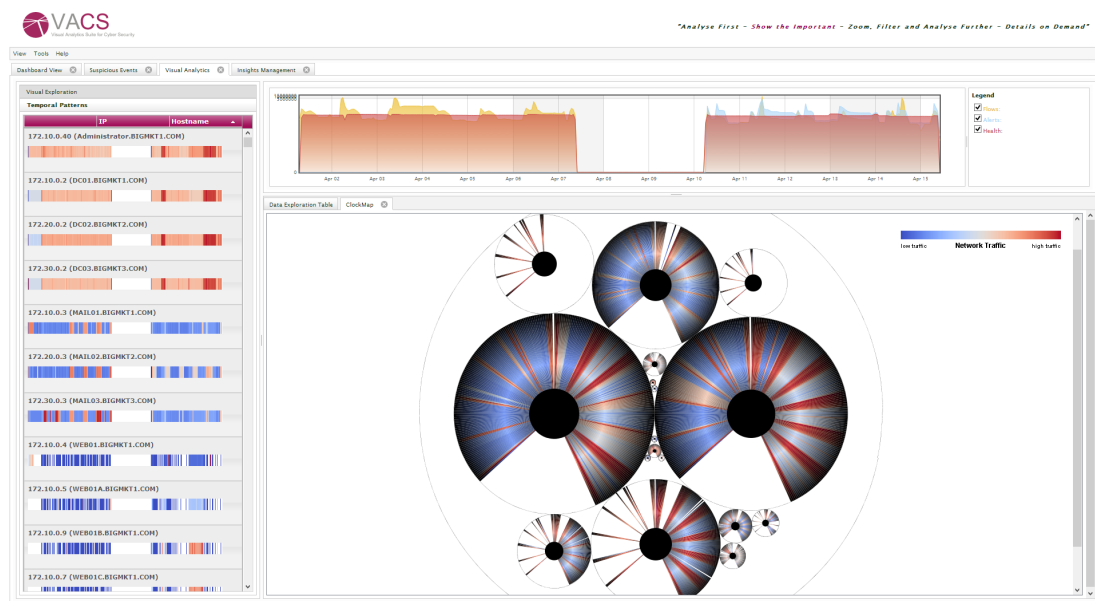
▲ **Figure 3.47** — DoS attack originated by internal hosts on 2013-04-14. The hosts continue to attack another external target on 2013-04-14 (Event 27).

### 3.4 Conclusions

In this chapter, we addressed three important use cases to support network activity analysis, which is a major factor to gain situational awareness for cyber security with respect to local computer networks. We designed, implemented, and extensively evaluated three prototypes, namely *VACS*, *IAS-Explorer*, and *ClockMap* to specifically address the limitations of related systems for internal/external, port activity, and host/server monitoring. We showed in particular, that the novel and scalable visualization technique *ClockMap* helped to visually explore large number of network hosts within their respective contexts.

Furthermore, *ClockMap* is generic enough to be used for other hierarchical time-series data as well. The technique combines a circular nested treemap layout with a radial glyph representation for time-series data and proved to be effective for comparative tasks on large amounts of hierarchically structured time-series data. When being used in combination with circular glyphs, the shape preserving property of circular nested treemaps outweigh the known disadvantages of such treemap variants and facilitates comparative tasks (especially for temporal peak detection tasks) within and across hierarchy levels.

Successful active participation in international competitions with *VACS* and the *BANKSAFE* system featuring the *ClockMap* approach helped to evaluate the advantages and limitations of our methods compared to a wide variety of other approaches. Eventually, the integration of *ClockMap* into *VACS* helped to combine the strengths of all integrated visualization techniques as seen in Figure 3.48, in which we even integrate the possibility to use the hierarchical *ClockMap* representation to visualize hierarchy clustering results of the time-series. The circles on the upper levels in Figure 3.48



▲ **Figure 3.48** — Integration of *ClockMap* into *VACS*. The integration combines the strengths of all visualization techniques into a single web-based visual analytics suite.

actually contain all network hosts having similar time-series. Therefore, the network can not only be represented with the focus on the organizational or subnet hierarchy, but also with respect to the actual behavior of the underlying hosts.

Overall, Chapter 3 primarily fulfills research objective RO1 as identified in Section 2.3, to introduce novel visual techniques for context-aware exploration to support visual analytics for network activity.





*“People often represent the weakest link in the security chain and are chronically responsible for the failure of security systems.”*

— Bruce Schneier

C H A P T E R



# 4

## Visual Analytics for Network Threats

### Contents

---

<b>4.1</b>	<b>Visual Overview for Attack Patterns</b>	<b>104</b>
4.1.1	Usage of Temporal MDS Plots for Attack Patterns	106
4.1.2	Evaluation using Network Security Case Study	107
4.1.3	Conclusions and Limitations	109
<b>4.2</b>	<b>Visual Correlation for Routing Anomalies</b>	<b>110</b>
4.2.1	VisTracer – Visual Analytics for BGP Prefix Hijacking	114
4.2.2	Evaluation using Case Studies	120
4.2.3	Conclusions and Limitations	126
<b>4.3</b>	<b>Visual Analysis for Malware Behavior</b>	<b>127</b>
4.3.1	Taxonomy of Visualization Systems for Malware Analysis	128
4.3.2	Conclusions and Limitations	133
<b>4.4</b>	<b>Visual Exploration for Attack Attribution</b>	<b>135</b>
4.4.1	Data Analytics for Threat Intelligence	137
4.4.2	Integrated Visualizations for MDC Exploration	137
4.4.3	Evaluation using Field Experiment	138
4.4.4	Conclusions and Limitations	142
<b>4.5</b>	<b>Conclusions</b>	<b>143</b>

---

NETWORK threats are an ever increasing challenge in computer networks and more general in the whole cyber world, having critical effects on the real world. While the previous chapter focused on monitoring use cases, how visual analytics can help to achieve situational awareness for internal/external, port activity, and host/server monitoring to recognize symptoms for possible threats, this chapter has a different motivation: To further advance situational awareness in cyber security, it is inevitable to analyze the behavior and eventually the possible impact of threats. This mainly refers to the *comprehension* and *projection* stage of situational awareness as introduced

in Chapter 2 (Figure 2.3), in which *threat analysis* is a major type of analysis. This chapter, therefore, focuses on use cases related to the identification and analysis of specific (i) attack patterns, (ii) routing anomalies, (iii) malware behavior, and (iv) attack attribution to relate characteristics of a threat to the ecosystem (threat landscape) on a larger scale. Eventually, this also helps to understand the impact or distinguish between the motivation behind a cyber security threat.

Section 4.1 focuses on providing visual overviews for attack patterns, while we make use of *Temporal MDS Plots (TMDS)* to address existing limitations of previous work, focusing mostly on IDS alerts, and are not generic enough to identify and visualize events based on arbitrary features. Section 4.2 focuses on a visual analytics system to visualize routing anomalies to identify BGP prefix hijacking, which has severe consequences on the whole cyber infrastructure. Section 4.3 gives an overview for the emerging field of visual analytic techniques for malware behavior. The literature review also revealed, that most of the reviewed attack pattern visualization do not really focus on the projection stage for situational awareness. In the context of visualization for network threats, this means to attribute and relate occurring attacks to the overall threat landscape to place them in the context of a larger campaign or attribute them to a specific attacker group. This attack attribution use case is discussed in Section 4.4.

## 4.1 Visual Overview for Attack Patterns

According to Shiravi et al. [216] the focus for visualizations supporting the identification and analysis of attack patterns is “*not only the detection of attacks but also the display of multistep attacks. Different types of attacks show different behaviors and accordingly different visual patterns appear*” [216]. Because it is hard to convey such information in a textual or tabular way, it is very natural to use visualization within such use cases. This is also the reason, why this use case comprises the largest body of research according to our literature review in Chapter 2 (Table 2.4).

### Related Work

Table 4.1 represents related work for attack pattern visualization methods. The table highlights the strong body of research focusing on this particular use case. Interestingly, most methods involve only a very limited number of data sources. Since 2009, more and more tools also include other data sources. However, still most attack pattern visualizations rely on IDS/IPS alerts together with packet traces and network flows.

From the perspective of used visualization techniques almost all widely used visualization techniques have been employed in the field of attack pattern visualization, while in recent years, node-link diagrams, timelines, and radial visualizations are slightly more frequently used.

Many approaches (e.g., [7, 47, 166, 81, 165, 164]) use treemaps to give a general overview of attacks, however, this often limits the possibility to convey the temporal aspects of an attack directly within the visualization. Other systems use alternative techniques to focus better on the temporal development and use scatter plots (e.g., [273, 50, 198, 142]), or pixel-based visualizations (e.g., [37]). *SnortView* [142], uses a 2D time diagram, similar to a scatter plot, in which the x-axis is mapped to time and the y-axis is mapped to the various source IP addresses (e.g., attackers). The various alerts are mapped to colored icons (glyphs) placed at the corresponding position in the scatter plot, representing the different attack types.

▼ **Table 4.1 — Related work for attack pattern visualization methods.** Overview of related work with respect to data source and visualization type.

Method	Use Case			Data Source												Visualization			Year												
	Attack Patterns	Routing Anomalies	Malware Behavior	Attack Attribution	Packet Traces	Network Flows	IDS/IPS Alerts	Firewall Logs	Vulnerability Scans	Meta Data	System Metrics / Status Reports	DNS Logs	BGP Messages	Server Logs	File System Changes	Audit Trails	Webserver Logs	Database Logs	HoneyPot Logs	Spam / Phishing Mails	Malware Files	Behavior Logs	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display			
Girardin [99]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	1999		
NIVA 2002 [180]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	2002	
NIVA 2003 [209]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	2003	
SnortView [142]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2004	
Teoh et al. [244]	✓	✓	-	-	✓	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	2004	
IDS RainStorm [2]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2005	
IP Matrix [143]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	✓	-	-	2005	
VisAlert [161]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	✓	-	-	2005	
Rumint [145]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	✓	-	2005	
VisAlert 2005 [160]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2005	
Visual Firewall [155]	✓	-	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	2005	
IDGraphs [198]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	2005	
PCAV [45]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	2005	
VisAlert 2006 [91]	✓	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2006	
Ren et al. [199]	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	✓	-	✓	2006	
Rumint [50]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	2006	
Xiao et al. [273]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2006	
Mansmann et al. [165]	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	2007	
SVision [182]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2007	
SpiralView [22]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2007	
Mansmann et al. [164]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	2007	
NFlowVis [81]	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	2008	
Mansmann et al. [166]	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	✓	2009	
VAssist [101]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2009	
Choi et al. [46]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2009	
MOVIH-IDS [119]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2009	
Yelizarov and Gamayunov [276]	✓	-	-	-	-	-	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	2009	
Chu et al. [47]	✓	-	-	-	-	-	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	✓	2010	
ENAVis [159]	✓	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	-	2010
Avisa [215]	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2010
PeekKernelFlows [260]	✓	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	-	✓	-	-	-	2010
Corchado and Herrero [53]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	2011
Jajodia et al. [130]	✓	-	-	-	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	-	2011
DAEDALUS-VIZ [125]	✓	-	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	2012
Zhao et al. [285]	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2012
Sol [30]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	-	2012
Alsaleh et al. [7]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	✓	-	✓	-	-	-	✓	2013
NetSecRadar [289]	✓	-	-	-	-	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2013
P3D [179]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	2013
IDS Radar [286]	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2013
OCEANS [41]	✓	-	-	-	-	✓	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2014
IMap [92]	✓	-	-	-	✓	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	-	2014
MVSec [287]	✓	-	-	-	-	✓	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	✓	-	-	2014
Toa [183]	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	-	-	2015
SNAPS [37]	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	-	2015

Such view provides a good overview of alerts, or even multi-step attacks of individual attackers over time. If one attacker conducts multiple attacks over time (and can be identified using the IDS system), the temporal evolution of the attacks becomes visible as timeline. However, in larger computer networks, a vast amount of external IP addresses are involved. SnortView can only provide very limited scalability, because the occupied screen space directly correlates with the analyzed time span and the number of external IP addresses, which can be quite high. Additionally, there is no integrated analytical support, to group or cluster various events to make usage of space more efficient. To address such limitations, we focus on the challenge to identify patterns based on events (or NetFlow records) over time. We analyze them and use multi-dimensional scaling (MDS) to place each event to a specific location based on an arbitrary number of weighted event features. In contrast to *SnortView's* approach, our layout won't necessarily need more space, when the number of external IP addresses increases.

#### 4.1.1 Usage of Temporal MDS Plots for Attack Patterns

This section builds on the following publication [133]<sup>1</sup>:

D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):141–150, 2016. ISSN 1077-2626. doi:10.1109/TVCG.2015.2467553 [133].

Together with Jäckle et al. [133] we published a novel technique to visualize patterns over time. This approach is also suitable for attack pattern visualization in the context of cyber security. Because I was primarily just involved in *evaluating* the approach, I quote the following brief description of the *TMDS* technique of our joint publication. For more details, the reader is referred to the detailed description of concepts discussed in Jäckle et al. [133].

*“Temporal Multidimensional Scaling (TMDS) [...] takes temporal multivariate data into account and visually presents the data enabling analysts to identify patterns and explore the data space [...] TMDS applies a sliding window approach on the data and computes a one-dimensional (1D) MDS for each window. The resulting sequence of 1D MDS mappings are then organized along the temporal axis: The x-axis represents the time, and the y-axis represents the MDS similarity value. Similar events are grouped over time and can efficiently be identified. To analyze the multivariate nature, we augment the visualization with a sequenced diversity matrix aligned with the MDS plot revealing the different temporal behaviors of single variables. Furthermore, we introduce a new algorithm to find similar patterns based*

<sup>1</sup> The publication's main contribution (TMDS) was developed by Dominik Jäckle, so most of it was also described by him in our joint publication [133]. I applied his technique to a real-world dataset as a case study and conducted a ground truth evaluation using VAST Challenge 2013. The responsibilities for this joint publication were divided as follows: Dominik Jäckle did most of the writing concerning the technique. I primarily focused on writing the case study and ground truth evaluation. Tobias Schreck actively contributed to the paper, while Daniel Keim gave advice and suggestions. All authors were also involved in proofreading.

*on the user selection and the behavior along dimensions. TMDS enables the efficient detection of recurring patterns and further allows to identify evolution of patterns, being based on varying scales and intervals. [...] Most existing approaches employ two-dimensional MDS projections. TMDS relies on 1D MDS projections, taking the second dimension in the plot to show the change of multivariate data patterns over time.” [133]<sup>2</sup>*

Figure 4.1 shows our *TMDS* application for a real network flow dataset. The main display, which looks like a large scatter plot is based on many vertically-aligned 1D MDS projections (as columns), which are placed next to each other. This means that all colored dots, having the same x-value, represent the events occurred in a single time window. The y-value represents the MDS similarity value.

During *calculation* of a single 1D MDS, events of the previous time windows are included as well. Consequently, highly similar events will be placed on a similar y-position in the respective columns. For example, most events visualized as blue-colored dots in (B) of Figure 4.1 represent NetFlow records, which have the same source IP address together with few specific TCP port numbers.

Previous work in network security visualization typically focuses either on general, temporal independent, patterns (e.g., [81]), or on temporal patterns (e.g., *TNV* [102]), which can typically not be analyzed promptly due to scalability and level-of-detail issues. At first sight, *TMDS* might look similar to *PortVis* [173], yet our approach does fundamentally differ as described in the previous paragraph. *PortVis* solely uses time and port range as axes to represent the events and thus particularly focuses on port scans. Our approach does not only focus on ports, but takes arbitrary (weighted) dimensions into account, and is therefore able to identify today’s complex temporal attack patterns showing general behavior. *TVi* [28] for internal/external monitoring also operates on temporal slices using entropy, but uses PCA-based techniques to analytically identify anomalous behavior using a timeline visualization combined with histogram charts.

To demonstrate the effectiveness of our method, we present in the following section a case study using network traffic. The interested reader, can find an additional evaluation using the VAST Challenge 2013 dataset in Jäckle et al. [133].

#### 4.1.2 Evaluation using Network Security Case Study

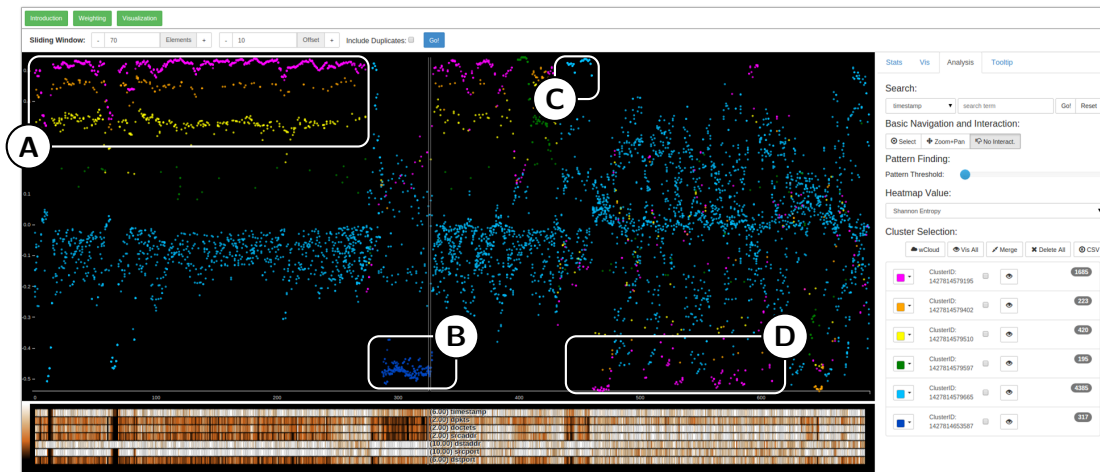
In this case study, we focus on all *loud* events of a full period of 24 hours of a public /16 computer network. We want to obtain a rough image of interesting events with different characteristics. The data is based on a privacy-preserving and anonymized data collection infrastructure, which we developed and used in previous research [81].

To facilitate this analysis, we use *Apache Spark*<sup>3</sup> to preprocess and sample the NetFlow data files, which are about 4 to 10 GB per day, to generate a suitable CSV file of incoming data flows only. This preprocessing step reduced the network flows to 16,474 records. Focusing solely on TCP traffic leads to 6,908 records as visualized in Figure 4.1. After loading the CSV file into our interactive prototype, we weight the different main dimensions with respect to increasing the impact of the IP addresses and the destination port. Because we analyze incoming network traffic, we are particularly interested in how possible attackers access services within our network. In such cases, the source port

<sup>2</sup> This paragraph was mostly written by Dominik Jäckle in our joint publication [133].

<sup>3</sup> <https://spark.apache.org/>

is less helpful to distinguish between different attack patterns, because it is assigned by the operating system or router from an ephemeral port range, respectively. However, the destination port is relevant to assign attacks to similar attack vectors. A higher weight of such ports leads to visual clusters of attacks to the same service (e.g., focusing on port TCP/80, which is the default port for HTTP traffic). After weighting the dimensions, *TMDS* is computed within seconds and the visualization display is loaded. It is quite usual for network traffic, that most connections are quite diverse and are hard to distinguish, because legitimate traffic does often not represent any clear patterns or clusters. We observe the same situation here, thus many records are diversely spread on the vertical axis (light blue dots) as seen in Figure 4.1. However, several interesting and unexpected visual patterns are clearly visible. We discovered various salient visual patterns using *TMDS*, which are labeled from A to D in Figure 4.1. Finding these events with diverse characteristics without visual support of *TMDS*, would require the analyst to issue various manual queries on the data. Manual queries would have been hard without knowing them beforehand and would also have been quite time-consuming. In the following a list of main patterns found with *TMDS*.



▲ **Figure 4.1** — Temporal MDS plots applied to network traffic data. For each temporal MDS plot (top) the sequentially aligned matrix (bottom) provides an overview of correlations among dimensions. The visualization reveals the attack patterns for a distributed brute-force attack (A, D) and various different port scans (B, C). *Reprinted from [133]. © 2016 IEEE.*

- **Pattern A: Distributed Brute-Force Attack** – This pattern reveals a long-term distributed brute-force attack from a distributed bot network on port TCP/22 to break into reachable SSH servers. Using the similarity clustering after manual selection of an arbitrary part of the pattern, it becomes obvious that the attack is operated over a longer period of time. Figure 4.1 shows all events in magenta related to this specific distributed botnet attack.
- **Pattern B: Massive Port Scan** – Drilling-down the visual pattern (highlighted in dark blue) reveals a massive port scan from a *single* IP address to a specific exclusive set of ports (TCP/80, TCP/81, TCP/443, TCP/8000, TCP/8080) of

many internal computers from a single external attacker, which is not related to the ongoing brute-force attack. The scan was operated from 10:36 until 10:56. The goal of this scan was to check for running web servers on various common ports.

- **Pattern C: Single Port Scan** – This pattern reveals a port scan to our network looking for accessible webservers on port TCP/80. In addition, some port scans search for open SMTP server on port TCP/25, which is typically performed to identify mail servers. Open mail servers can be used as open relay for sending spam.
- **Pattern D: Brute-Force Continuations** – The magenta color refers to the same characteristics as seen in Pattern A. Some attackers are still trying to attack SSH services, however in a much more subtle way than during night time as seen in Pattern A.

### 4.1.3 Conclusions and Limitations

In previous years, there was a lot of research focusing on visualizations for attack patterns. However, most of them relied on IDS alerts to provide good high-level overviews of major network threats, which are previously identified using mostly signature-based intrusion detection systems. However, this also limits the possibility to provide a holistic view for more complex targeted attacks, primarily for advanced persistent threats (APT), in which possible traces in many different data sources provide important hints contributing to a thorough threat analysis. Our approach using *TMDS* plots is more generic and takes into account an arbitrary number of weighted features and combines analytics with a scalable visualization to identify and reveal temporal attack patterns. Compared to other approaches, the visualization is independent from specific features, because the similarity of events is highlighted to identify re-occurring or orchestrated attacks sharing common features. In our case study, we showed the usefulness of this approach in the analysis of network traffic and identified various interesting findings.

However, there are various limitations of *TMDS*, which will have to be addressed by future research. The technique depends on various parameters. Because these parameters depend on the data characteristics and size, it is hard to predict and set them automatically. Eventually, the user is responsible for setting an appropriate window and step size to define the sliding windows. Furthermore, the current implementation of *TMDS* needs a lot of memory and processing time. Therefore, we had to heavily sample the original network-security datasets, so that they could be analyzed with *TMDS*.

## 4.2 Visual Correlation for Routing Anomalies

The sections coming next build mostly on the following publications [25, 84]<sup>4</sup>:

F. Fischer, J. Fuchs, P.-A. Vervier, F. Mansmann, and O. Thonnard. *VisTracer: A Visual Analytics Tool to Investigate Routing Anomalies in Traceroutes*. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 80–87, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379701 [84].

E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P.-A. Vervier. *Visual Analytics for BGP Monitoring and Prefix Hijacking Identification*. *IEEE Network*, 26(6):33–39, 2012. ISSN 0890-8044. doi:10.1109/MNET.2012.6375891 [25].

In the previous sections, we focused on attacks on the Internet and often assumed that the IP addresses involved in an attack are meaningful. An active brute-force attack originating from a specific IP address most likely means, that the respective host is either owned by the attacker, or is actively misused without the knowledge of the legitimate owner (e.g., a host which is part of a botnet). On the other hand, if we receive an e-mail from a legitimate mail server or legitimate IP address, which is not blacklisted, the e-mail is most likely less suspicious than others.

However, these assumptions only hold, when the underlying routing is correct. Routing is a fundamental concept in the Internet. Correct path announcements are important to reach the correct destination servers. Despite of the importance and the severe consequences of routing issues, the responsible border gateway protocol (BGP) is quite vulnerable. Announcing malicious routing paths can be used to hijack IP blocks. As a result the attacker can conduct malicious activities from legitimate IP addresses. Distribution of vast amounts of spam is a scenario where the misuse of legitimate IP prefixes helps the attackers to circumvent widely used IP-based blacklists.

### Related Work

In Table 4.2, we provide an overview of visualization method for BGP-related analysis tasks. These systems can be classified into various main categories, which will be described in the following.

### Systems with High-Level AS Overviews

*BGPlay* [18], which is one of the most popular BGP analysis tools, uses a node link diagram to present an intuitive high-level AS view to show the autonomous systems and their connections with each other. *BGPlay* was improved by integrating a topological

<sup>4</sup> The responsibilities for our joint publication about *VisTracer* [84] were divided as follows: Johannes Fuchs and I did the programming and the writing. Pierre-Antoine Vervier provided the data and was also involved in the writing, especially with respect to the case study. Florian Mansmann and Olivier Thonnard did the proofreading and gave advice. The joint publication [25] was an outcome of our close collaboration with BGP experts in the VIS-SENSE project, providing a survey of BGP visualization methods to identify prefix hijacks, in which I mainly contributed in describing the various tools.



▼ **Table 4.2 — Related work of visualization methods for routing behavior.** Overview of related work for visual analysis of routing behavior and anomaly detection.

Method	Use Case				Data Source														Visualization					Year					
	Attack Patterns	Routing Anomalies	Malware Behavior	Attack Attribution	Packet Traces	Network Flows	IDS/IPS Alerts	Firewall Logs	Vulnerability Scans	Meta Data	System Metrics / Status Reports	DNS Logs	BGP Messages	Server Logs	File System Changes	Audit Trails	Webserver Logs	Database Logs	HoneyPot Logs	Spam / Phishing Mails	Malware Files	Behavior Logs	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display	
Teoh et al. [242]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2002
Teoh et al. [243]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	✓	-	-	✓	✓	2002
ELISHA [241]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	2002
BGPlay 2003 [18]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2003
EventShrub [245]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	2004
Teoh et al. [244]	✓	✓	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	✓	-	-	✓	✓	2004
BGPlay 2005 [49]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2005
TAMP [271]	-	✓	-	-	-	✓	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2005
BGPlay 2006 [55]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2006
Link-Rank [148]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2006
VAST [181]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2006
BGP Eye [246]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	✓	-	-	✓	-	2006
BGPeeP [214]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2008
Papadopoulos et al. [186]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2012
BGPfuse [187]	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	-	2013

map [55] to represent hierarchies. Both tools provide timelines, which can be used to focus on interesting time intervals. Animation helps to present routing changes and route flappings. Several colored lines describe the advertised routes as node-link diagram to the selected prefix.

While this interactive *animated* visualization is quite intuitive for the visual exploration of historic events in BGP data, the analyst must have a clear idea of which time span and which prefix is relevant for the analysis. Compared to static representations, animation is time-consuming and the analyst needs to focus on many changing aspects of the graph. The main benefit of such an animated view is to present a known case, but not necessarily to identify a suspicious event.

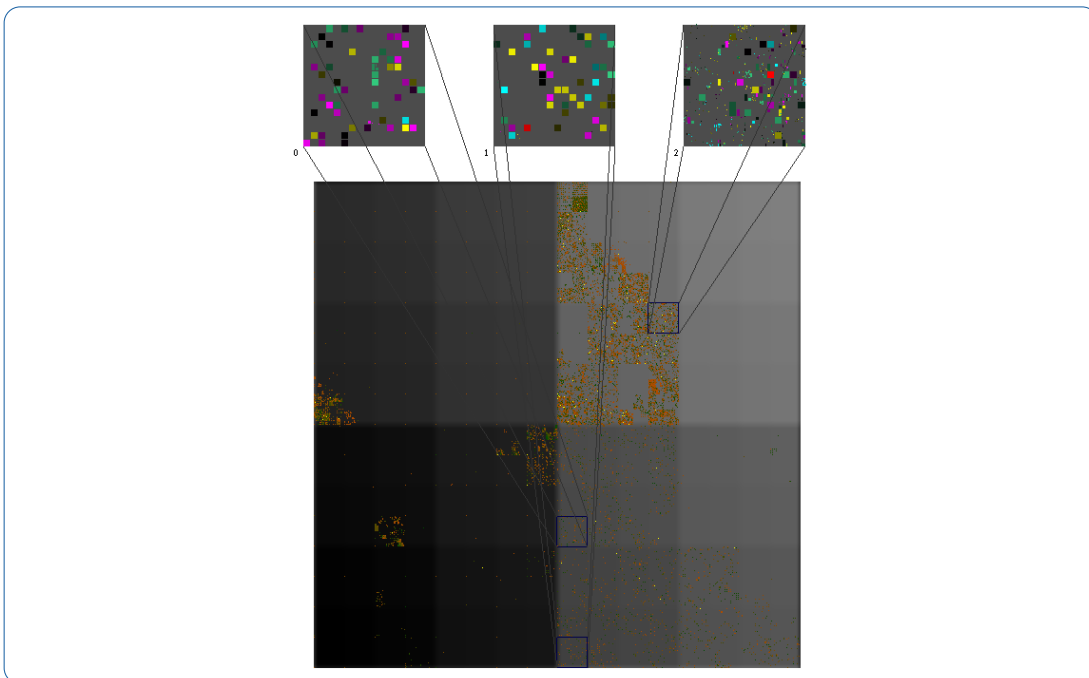
*Link-Rank* [148] is a similar system, because it also uses a graph based representation of the ASes. Additionally, the edges are weighted according to the number of routes and changes between the different AS links. With this supplementary information the analyst can observe routing changes and link instabilities. Activity plots further help to focus on the most suspicious update bursts, which might indicate prefix hijacking resulting in major route changes.

Another tool, focusing on animated node link diagrams, is *TAMP* [271]. It displays a pruned graph for the network topology, an animated clock with controls to show and manipulate the time of the current state of the graph and another detailed chart to present the events belonging to a selected edge. Compared to other tools, strong statistics are included to detect correlations between BGP events at any time scale.

The algorithms can be enriched with additional data sources like traffic flows or router configuration files to improve the diagnosis of BGP anomalies. The combination of statistical methods, data enrichment and visualizations helps to detect prefix hijacking, route flapping and anomalies in long time periods.

*VAST* [181] uses 3D visualizations to show topological connectivity between different ASes. Interaction possibilities like rotating, zooming, or panning help to explore the 3D space. Furthermore, different filter techniques provide the possibility to focus on certain aspects of the data. The tool allows mainly to visually explore AS connectivity and to identify critical infrastructures. Within the visualization update bursts of specific ASes become visible, which can be an indicator for occurred prefix hijacking.

*EventShrub* [245] uses an automatic anomaly detection algorithm combined with a tightly coupled visual timeline. Pie charts used as small glyphs are plotted to this timeline to represent the different instability events. This representation helps to identify deviations from normal behavior.



▲ **Figure 4.2** — Pixel-based visualization of *ELISHA*. The main visualization consists of a scalable pixel-based approach to display BGP data. Each pixel represents an IP address with a color encoding according to the corresponding BGP event. The three detailed windows at the top enlarge areas of interest to better analyze single IP addresses. *Reprinted from [25]. © 2012 IEEE.*

### Systems with Low-Level IP Views

In contrast to the previous tools, which focus on high-level AS overviews, there are various tools focusing on the low-level IP space. *ELISHA* [241] makes use of a pixel-based approach. The screen is filled with colored pixels, each representing single IP addresses. They are laid out according to their corresponding IP range. The BGP messages are classified in different event types. Visually mapping this information to color provides a scalable and space-filling overview visualization as seen in Figure 4.2. With this

animated visualization, analysts are able to detect, explore, and visually present routing anomalies and MOASes<sup>5</sup>. Overall, *ELISHA* provides an IP prefix centered approach without representing the overall AS routing paths.

An overview visualization focusing on textual content instead of temporal aspects is provided by tag clouds, which are a key component in *BGPeeep* [214]. The different tags represent the names of autonomous systems. The size of tags depends on the number of update messages for the specific AS. To make use of the hierarchical structure of IP addresses and to provide a more IP-space centered view, horizontal parallel axes are used by *BGPeeep*. The first axis represents the AS number; the other four, one byte of an IP address. An update message is represented by a line intersecting the axis at appropriate positions. Using this visualization technique, it is possible to reveal potential router misconfiguration, route flapping, or multiple advertised prefixes.

### Systems with Multiple Views

*BGP Eye* [246] combines data mining techniques and visualizations using multiple views. Update messages are classified and clustered. An overview visualization displays the activity among different ASes in a graph layout. Additionally, a 3D matrix with connecting lines, reveals more detailed information about a single AS. Therefore, prefix hijacking or changes in the overall routing behavior can be detected.

*RIPEstat*<sup>6</sup> is a popular web interface, which is continuously improved, with a variety of different charts to show historic activities or different distributions related to the selected AS or IP prefix. These visualizations do not present a general overview to detect anomalies, but help to investigate individual cases, and also includes various techniques found in academic research papers (e.g., integrated version of *BGPPlay* [18]).

### Other Methods and Approaches

Besides the visualizations of routing data, several solutions to *secure* BGP have already been studied in by Bush and Austein [35] and Kent [137], but the high computational cost of using cryptography and the required changes in protocol and infrastructure retain their deployment. Automated BGP hijack detection techniques attempt to uncover abnormal changes in the routing infrastructure likely due to a BGP hijack by monitoring the *control plane* and/or the *data plane*. Most automated systems [147, 194] only monitor BGP updates and trigger an alert when a new advertisement conflicts with their model of the Internet topology. Various other methods [15, 120, 283, 288, 236] also use data plane information to collect information about the different hosts and networks along the forwarding path from a source to a monitored network. Several features of data plane traces can be leveraged to help detect abnormal routing changes, e.g., a network reachability change [236], an AS-level traceroute deviation [283], a significant change in the traceroute path length [288]. Finally, Hu and Mao [120] combine control plane BGP hijack detection techniques with host fingerprints.

However, these automated techniques do not really enhance situational awareness with respect to BGP anomalies, because the context is not conveyed to the analyst. Therefore, visualization techniques are needed to visually explore the situation to distinguish between false positives and actual suspicious anomalies.

<sup>5</sup> Multiple Origin Autonomous Systems: IP prefixes appearing to originate from more than one AS. Such originating AS is called MOAS.

<sup>6</sup> <https://stat.ripe.net/>

Overall most of the aforementioned visualization tools show the routing changes mainly as animation, which is appropriate for visually presenting a particular known event. For exploratory analysis, animation is not entirely satisfying. Therefore, other techniques have to be investigated in order to improve the temporal analysis. To combine the strengths of scalable and informative approaches for long-term analysis, the tight coupling of different techniques seems to be promising.

Our approach, which will be presented in the following sections, instead makes use of a combination of pixel-based techniques to present anomalous events in an overview, and glyph-based techniques to represent historical information for analyzed targets. We do also include a graph representation. However, our focus is the direct integration of temporal information into the nodes of the graph using a temporal glyph representation. Besides of the optional animation, this static integration in our approach can help the analyst to get a quick overview of the path without having to replay the whole communication as animation to understand the temporal changes.

In our system we leverage different features of the traceroutes like the IP/AS paths (based on the BGP messages), the route length, the host and AS reachability as well as some BGP information to detect abnormal routing changes. We also correlate them to help determine whether observed routing changes are benign or malicious.

#### 4.2.1 VisTracer – Visual Analytics for BGP Prefix Hijacking

The focus of our work is the large-scale analysis and exploration of routing anomalies for IP addresses starting to send spam in the Internet. This is achieved by actively tracking and measuring the traceroutes to the origin IP addresses over longer periods of time to eventually monitor possibly malicious path changes. Because of the vast amount of trace data with their changing underlying BGP routes, it is not helpful to just visualize the raw data. To make sense of the data it is important to algorithmically identify anomalies first. The tight integration of visual displays can be used to get an overview for quick ad-hoc analyses to identify noteworthy events and to differentiate them from false positives. The proposed visualizations in our work help to visually correlate anomalies, gain deep insights, and explore the events within their context of historic and related anomalous traceroutes. Furthermore the analysts can push their findings back to the system. This feedback could then be used for further improving the underlying anomaly detection algorithms.

The three main contributions of our work with respect to BGP anomaly detection are (i) a visual analytics tool called *VisTracer* to analyze large-scale traceroute data, (ii) the integration into a large-scale analysis system and (iii) novel glyph- and graph-based summary visualizations for traceroutes. Additionally, we present an in-depth discussion of recent case studies for suspicious routing anomalies with respect to spam activities.

#### Analysis Infrastructure

Manipulating the Internet routing infrastructure to hijack a block of IP addresses involves modifying the route taken by data packets so that they reach the physical network of the attacker. A system called *Spamtracer* [257] has been developed by Vervier and Thonnard [257] to monitor the routes towards malicious hosts by performing `traceroute` measurements repeatedly for a certain period of time. IP-level routes are translated into AS-level routes using live BGP feeds. The motivation for monitoring data plane routes towards specific hosts involved in spam campaigns is to collect the route taken by data

packets to reach these hosts as soon as a spam is received from them. By performing multiple measurements on consecutive days for a certain period of time, typically one week, routes towards a given host or network can be compared and analyzed in depth to find evidences of a possible manipulation by an attacker of the routing infrastructure.

This system is based on a linear data flow where a feed of IP addresses to monitor is given as input and a series of enriched traceroute paths produced as output from which abnormal patterns can be uncovered. The incoming feed of IP addresses are retrieved from *Symantec.cloud* [232] spamtraps. This data is enriched with IP traceroutes. A customized version of the classic `traceroute` function is implemented and takes advantage of ICMP, UDP and TCP packets to increase the likelihood of hosts to be reached by them. Due to the many artifacts that can be found in IP-level traces, we also build the AS-level routes. The IP-to-AS mapping is performed using live and distributed BGP feeds from *RouteViews* [202] to obtain as accurate and complete mappings as possible. Additionally, information about the different hosts, AS owners, IP networks and geo locations is collected.

### Extracting Routing Anomalies

We analyze the collected routes to uncover abnormal routing changes and classify them as benign or malicious. Routing anomalies are extracted independently for every monitored IP addresses. The first approach does focus on extracting routing anomalies from BGP hijacking scenarios, while the second one searches for suspicious patterns based on different metrics. To identify malicious BGP hijacks, we start from existing scenarios of BGP hijacking [120] for which we know the resulting routing anomalies. However, it has to be considered that such routing anomalies can also result from benign BGP routing practices, e.g., multi-homing of customer ASes by ISPs, or from non-malicious incidents due to misconfiguration or operational errors.

- **Prefix Ownership Conflicts** occur when a block of IP addresses appears in the Internet routing infrastructure as originated by multiple ASes. This routing behavior can be the result of a hijacker advertising someone else's IP space in order to attract traffic to or originate traffic from that IP space. Advertising the *same prefix* is a possible way for BGP hijacking, if the IP prefix is already advertised by a different AS. This technique creates a routing anomaly referred to as Multiple Origin AS (MOAS). Announcing a slightly *different prefix* can also be used for tampering the ownership of a given IP prefix, which can be more (resp. longer) or less specific (resp. shorter). In this case, we refer to this anomaly as a Sub Multiple Origin AS (subMOAS).
- **BGP AS Path Anomalies** occur, when the location of a network in the Internet AS topology changes. As a result of a BGP hijack it is likely that the sequence of ASes traversed from two different points will change. Significant changes in the BGP AS paths should be investigated to determine if they are indeed benign or if they result from a malicious manipulation of the routing infrastructure. The *Next-Hop AS* anomaly can be observed with a certain number of different next-hop ASes, i.e., ASes next to the origin AS in an AS path, for a given origin AS and BGP collector. A *Complete AS Path* anomaly consists in observing a significant change in the AS paths for a given origin AS and BGP collector.

The second approach searches for suspicious patterns in traceroutes based mostly on metrics already used in previous works [288, 236].

- **Traceroute Destination Anomalies** refer to suspicious values in features related to traceroute metadata. *Host/AS reachability* defines if a destination host or AS towards a given IP address is reachable (unreachable) for a certain number of days during the monitoring period and suddenly becomes unreachable (reachable) and remains like this until the end of the monitoring period. This reachability anomaly can result from a major routing change which causes the destination host or AS to become (un)reachable. The *hop count* or the length of a traceroute path is the value of the last TTL for which a reply to our probe IP packets has been received. The hop count anomaly is the consequence of a significant and sudden change in the hop count. This situation suggests that an important routing change occurred to permanently change the route taken by packets to reach the destination network.
- **Traceroute Path Anomalies** refer to suspicious changes in the sequence of hops traversed by traceroute paths to a given destination host. Using the different features collected for IP/AS hops, we can consider a traceroute not only as a sequence of IP addresses or ASes, but also as a sequence of countries, domain names, RIRs, etc. These alternate paths are leveraged in this detection of suspicious traceroute paths. The *AS-level Path Anomaly* consists in observing a significant change in the AS-level paths towards a given IP address.

*Country-level Path Anomalies* are observed by extracting traceroute paths towards a given host exhibiting significant discrepancies in the sequence of traversed countries. This assumes that the countries traversed to reach a given destination from a given source is likely to remain constant even if routing changes occur at the IP or AS levels.

## Design and Development Process

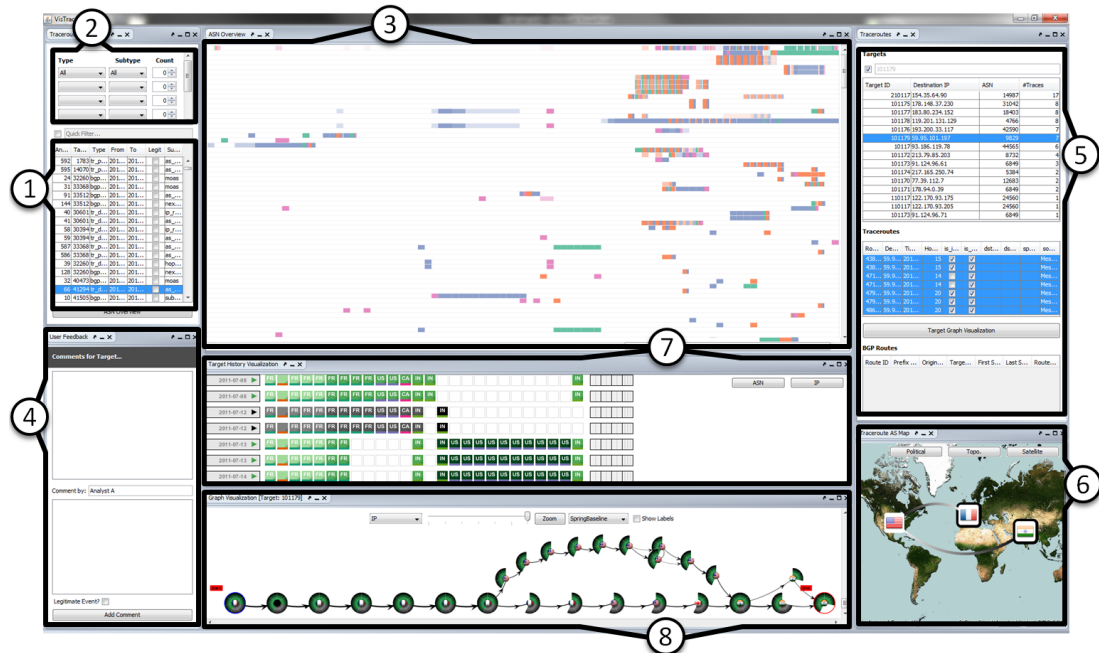
To bridge the gap between domain and visualization experts to generate a useful tool to solve the analyst tasks is a challenging problem. We addressed this issue through a tight collaboration with security experts. While my own background is in data visualization, but also with work experience in system administration, I could support the effective communication between these two groups. Such a situation, where one person has the role as a liaison has been formalized by Simon et al. [220] with the “*goal to overcome the interdisciplinary communication issue*” [220].

The development process of *VisTracer* also followed the general idea of co-creation [205], in which the target group, in our case the BGP security expert, was an active part of our team. This enabled us to incrementally design and develop *VisTracer* according to the specific needs of the BGP analysts.

## Graphical User Interface

The continuously growing *Spamtracer* database can be accessed by the analyst using our visual exploration tool called *VisTracer*. The graphical user interface is built in a way to satisfy the needs of experienced analysts by providing an overview linked to more detailed visualizations. This helps to solve the different analysis tasks. The individual

views can be placed according to the user’s preference or adjusted to the working environment which is important, when the tool is used in multi-display environments.






▲ **Figure 4.3** — Graphical user interface of *VisTracer*. (1) and (2) provide access to constraint filters and a table with observed anomalies. (3) Visual ASN Overview with occurred anomalies. A Feedback Panel is provided in (4) and access to individual traceroutes in (5) with map-based (6), glyph-based (7) and graph-based (8) visualizations. Reprinted from [84]. © 2012 ACM.

The general workflow of *VisTracer* is inspired by Shneiderman’s information seeking mantra of having the overview first and then focusing on certain areas of interest to retrieve additional details [218]. The overall graphical user interface is shown in Figure 4.3.

The left panel (1) provides a tabular anomaly view with all occurred anomalies. To investigate specific cases a filter box is integrated for quick ad-hoc queries. Using different constraints (2) for anomaly types and subtypes the user can focus on the different classes and combinations of anomalies. Based on the given constraints the *ASN Overview* (3) provides an overview of all anomalies using a visual representation. Findings can be stored in the database using the feedback panel (4), which can be used to annotate anomalies and comment on findings to make them accessible for other analysts. The right panel (5) provides tabular access to all destination targets with their traceroutes. Selecting entries in any of the tables will update the loaded visualizations for further investigation. A zoomable geographic map (6) to visually present the currently selected AS path is included. The *Visual Traceroute Summary* (7) is a compact visual representation, while the *Temporal Graph Representation* (8) is used to get an in-depth overview of temporal usage for involved nodes.

▼ **Table 4.3** — Various glyphs used in *VisTracer* visualizations. An overview about the three glyphs, which are incorporated in the various visualizations in *VisTracer*.

Glyph	Usage	Description
	<b>Anomaly Glyph</b> Visual ASN Overview	Used as glyph in matrix overview to represent most dominating anomalies for the various AS networks over time. Color shows the percentage distribution of various anomaly types.
	<b>Hop Glyph</b> Target History Visualization	Represent if the current traceroute hop is being used. Color at bottom and label represents country. Background color reflects incomplete (gray) and complete (green) traces, while brightness is hop's latency.
	<b>Clock Glyph</b> Temporal Graph Representation	Used as hop representation (including country flag) in node-link diagrams. Each segment represents a particular day, on which a traceroute was gathered. The color of a hop's segment shows if the node was part of the respective traceroute.

### Visual ASN Overview

The main starting point for an exploratory analysis is to monitor different ASes and the occurring anomalies over time. Therefore, a zoomable matrix layout has been chosen as the basis for the visual marks shown in Figure 4.3 (3). The x-axis encodes the time and the y-axis the different destination ASes of traceroutes. By default, the ASes are ordered according to the total number of anomalies, while other sorting algorithms might be more appropriate for finding common patterns and correlations between different ASes. Due to the fact that multiple anomalies of *different* types can occur on specific points in time, rectangular glyphs are used to encode this additional information. Glyphs have the advantage of showing multiple data dimensions in a space efficient compact way. Each glyph has a fixed size and consists of several colored vertical stripes. Each colored stripe encodes one type of anomaly. The stripe width is proportional to the amount of daily anomalies for the respective event type. We decided to chose this additional size encoding to emphasize on the most prominent anomaly types in the overview, especially when they spread over longer periods of time. The stripe's color encoding is based on a qualitative color scale provided by *ColorBrewer* [31] and helps to visually distinguish between the different kinds of anomalies. Therefore, ASes with characteristic colored patterns are a visual hint for reoccurring anomalies. To further focus on the “hot spots” with lots of anomalies, opacity is used to encode the overall number of occurred events. Table 4.3 shows a closeup of such a single anomaly glyph. An AS-based normalization is used to avoid artificially promoting heavily used large ASes. Suspicious ASes can be further investigated through double clicking on the different rectangles, which updates the different views and tables to provide more details on demand.



### Target History Visualization

Traceroutes to the same destination can be investigated in the *Target History Visualization* as seen in Figure 4.3 (7). The main idea of this visualization is to provide a visual traceroute summary to show hop usage variances of single traceroutes to the same target. Therefore, the x-axis encodes the individual hops and the y-axis the traceroutes on the different days. Whenever a hop is used within a traceroute a small glyph is placed accordingly. This rectangular glyph encodes the country code of the hop with a small label and a colored bar. With the help of this colored bar, connections within the same country can be spotted preattentively. The main color of the glyph reflects whether the traceroute was complete (green) or incomplete (gray). This prominent feature is visible at first sight because it is considered of high importance. Additionally, brightness is used to encode the latency of the individual hops. A closeup of this glyph can be seen in Table 4.3. At the end of each traceroute row, a small anomaly container is placed. The container represents the four main types of anomalies with equally sized rectangles. These rectangles are further divided into smaller rectangles representing the subtypes. Whenever a type/subtype combination can be found in a traceroute the corresponding rectangle is colored. Thus, anomalies lasting for a longer period can be easily detected as a reoccurring pattern over many traceroutes. Suspicious traceroutes with lots of anomalies show several colored rectangles and, therefore, are easy to spot. Examining the anomalies in combination with the used hops and the completeness of the traceroutes over time can lead to relevant findings and helps the analyst to understand the traceroutes. This visualization is especially effective to get an overview of the used hops in the different traceroutes.

### Temporal Graph Representation

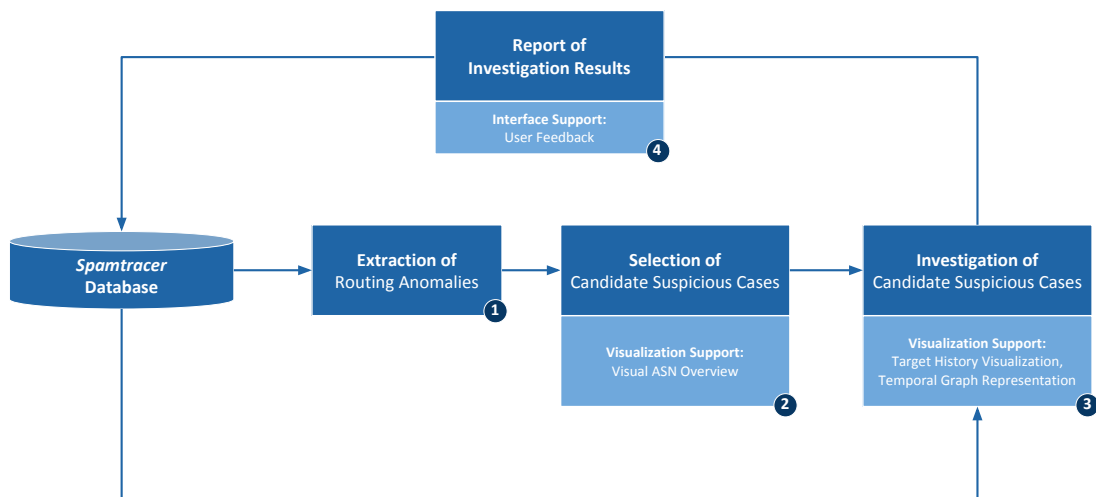
The previous visualization does not focus on following the exactly used routes or the identification of the most common route in the correct order. To solve this task, an additional graph visualization is provided as seen in Figure 4.3 (8). The graph layout is extended with an additional glyph encoding to show routing changes over time. The nodes represent the different hops, the edges show the connections with each other. The width of an edge depends on the amount of traces using this exact connection. The nodes are visualized by circular glyphs with equally sized slices and small flags reflecting the country of the hop as can be seen in Table 4.3. Because of the aspect ratio, the circular glyphs can directly be integrated into the graph nodes without wasting additional space for this temporal information or requiring disturbing and more time-consuming animation. The number of slices depends on the amount of traceroutes shown in the graph. The clockwise arranged slices represent the different traceroutes for the selected days. When a hop was used in a traceroute the respective slice is colored, otherwise it is not displayed at all. The color depends on whether the traceroute reaches its destination or not. This encoding supports the analyst in detecting the main route (i.e., based on the path's width), the usage of hops (i.e., the proportion of colored slices), the reachability of the destination (i.e., the hue of the colored slices) and the temporal development of the route (i.e., the partition of the slices). Additionally, the geographic location of the corresponding country can be taken into account in the layout to highlight possible route flappings between different countries with the help of the graph's layout. To focus on the main route, we additionally propose an *Enhanced*

*Baseline Layout* which displays the most common path at the bottom. The hops, not being part of the baseline are arranged in a force-directed way above the baseline.

Combining the different views or looking at them individually supports the user in the different analysis tasks. To evaluate the tool’s effectiveness, the following section describes the analyst’s workflow and how the visualizations help.

#### 4.2.2 Evaluation using Case Studies

In this section we describe how suspicious routing events are identified and how the *VisTracer* framework reflects this workflow to assist the analyst. We also present two case studies of routing events identified as suspicious using the developed visualization tool. Figure 4.4 depicts the steps involved in the analysis of the network traces collected by *Spamtracer*. Furthermore, this figure shows where in the workflow the visualizations can assist the analyst in examining the data. In detail the analysis is based on (1) automatically extracting routing anomalies from the traces, (2) selecting the monitored ASes having a meaningful set of anomalies, and (3) investigating cases using all the collected data to identify the suspicious cases. The result of the investigation of a case is finally reported back to the database (4).



▲ **Figure 4.4** — **Visual analysis workflow in *VisTracer*.** The figure shows the overall interactive analysis workflow relating the various steps to the visualizations and views integrated in *VisTracer*.

*VisTracer* supports the *Selection of Candidate Suspicious Cases* by providing a graphical user interface to filter for anomalies which match a given set of constraints on the type, the number and the time of appearance of the anomalies. These correspond to the most likely suspicious cases. This step is associated with the *Visual ASN Overview*, which allows the analyst to define the constraints on the anomalies and then explore the resulting set of targets aggregated at the AS level. The *Investigation of Candidate Suspicious Cases* means to investigate the suspicious cases with the help of the collected traces as well as some external routing information services to determine if a case is benign or if it results from a malicious BGP hijack. When investigating a case, the *Temporal Graph Representation* and *Target History Visualization* as well as the

traceroute hop list provide the analyst with all the data available to determine whether the routing anomalies observed reflect a malicious routing behavior. To communicate and further make use of the findings the tool also focuses on *Reporting of Investigation Results*. The feedback loop embedded in *VisTracer* allows to share the result of the investigation with other analysts.

The *Spamtracer* dataset used to produce the following two case studies contains traceroutes collected from April 2011 until the end of August 2011. 848,916 data plane routes were collected towards 239,907 IP addresses and 5,912 ASes. After the routing anomalies were extracted from the traces 41,430 destination IP addresses were found to have at least one anomaly. Given the high number of cases exhibiting at least one anomaly, we decided to focus on cases having the following combinations of anomalies:

- **BGP Origin & BGP or Traceroute Path Anomalies** – Select cases exhibiting a prefix ownership conflict with a significant change in the BGP or traceroute AS path.
- **BGP Origin & Traceroute Destination Anomalies** – Select cases exhibiting a prefix ownership conflict with either an IP/AS reachability change or a significant data plane route length change.
- **Traceroute Destination Anomalies & BGP or Traceroute Path Anomalies** – Select cases exhibiting a significant change in the BGP or traceroute AS path with an IP/AS reachability change or a significant data plane route length change.

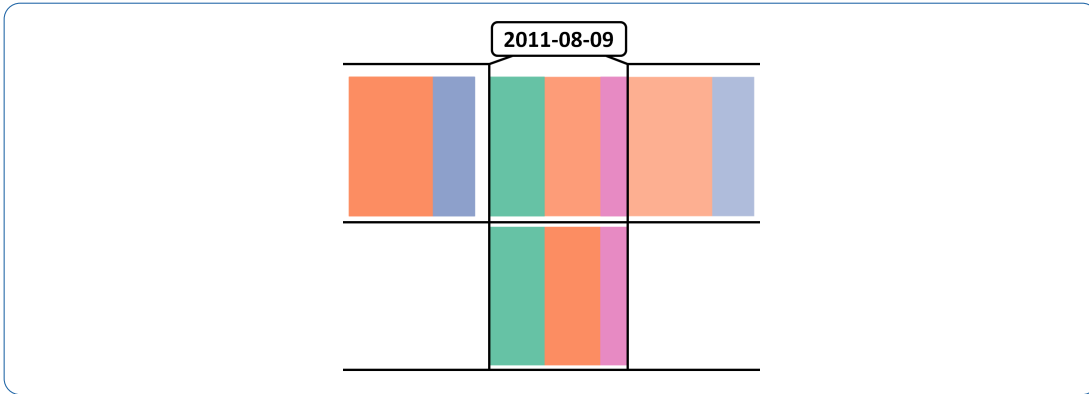
We have thus applied these filters in the *Traceroute Anomalies* panel of *VisTracer* to focus our analysis on these cases.

### Case Study: Analysis of Suspicious BGP Anomaly

The first case study conducted by our BGP security experts, presents the visual analysis of a network whose traffic was apparently hijacked by another AS. Actually, we show how such a case can be uncovered and investigated using the visualizations and other information provided by *VisTracer*.

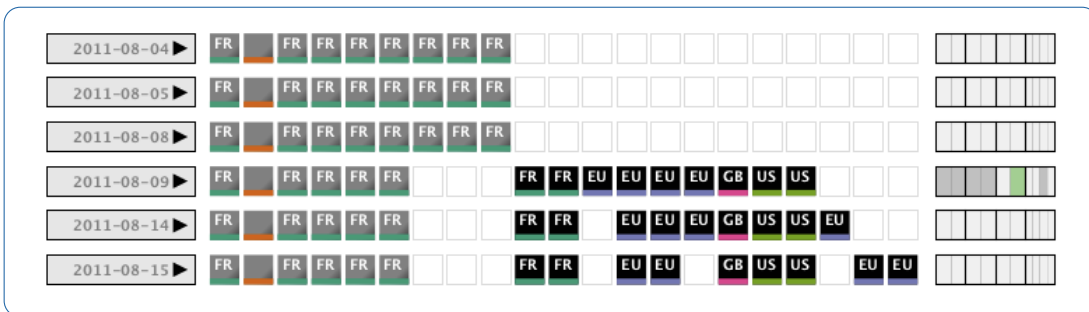
*“From the ASN Overview visualization, one particular case caught our attention, which can be seen in Figure 4.5. Two ASes actually appeared to share several anomalies, which occurred on the same day. The visualization allows to extract such time correlation between anomalies in different ASes thanks to the ASNs and time dimensions. Looking at the anomalies extracted for the two ASes reveals (i) a Traceroute Destination Anomaly (related to the destination AS reachability), (ii) Traceroute Path Anomalies, (iii) BGP Path Anomalies (AS Path Deviation) and, (iv) a BGP Origin Anomaly (related to a subMOAS conflict).*

*We can make use of the Target History Visualization to have a first view of the traceroute paths and the uncovered routing anomalies. Figure 4.6 shows the set of IP hops traversed by traceroutes from the vantage point in France to the destination host throughout the monitoring period. From this visualization we can say that there is a noticeable change in the set of traversed IP hops between the third and the fourth traceroute. The six*



▲ **Figure 4.5 — Suspicious AS networks.** Closeup of the *Visual ASN Overview* showing two nearly identical anomaly distributions for two different ASNs at the same point in time. *Reprinted from [84]. © 2012 ACM.*

*routing anomalies uncovered for these traceroutes on the fourth day confirm that a major routing change occurred. In this case, a change in the origin AS of the destination IP prefix occurred at the same time as a change in the sequence of ASes traversed both in the traceroutes and in the BGP AS paths. The BGP Origin Anomaly, in the third column, has been marked as benign (green) by Spamtracer, because the two conflicting ASes were found to have a provider-customer relationship.*

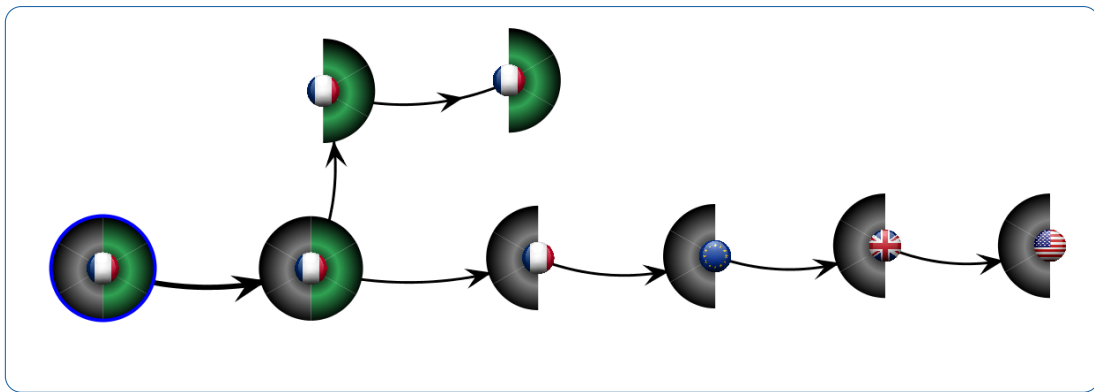


▲ **Figure 4.6 — Target History Visualization of the first case study.** The visualization shows the significant difference in the ASes traversed between the third and fourth day. The routing anomalies observed on the fourth day are also shown. *Reprinted from [84]. © 2012 ACM.*

*To further investigate the case, we make use of the Graph Visualization, which is presented in Figure 4.7 for the same monitored host. The Graph Visualization allows the analyst to look at the IP-, AS- or the Country-level traceroute paths, i.e., the sequence of IP hops, ASes or countries traversed. While the AS-level graph is particularly well suited to investigate abnormal changes in inter-domain routing, the IP- or Country-level graphs can also be leveraged to investigate routing anomalies. Actually, they are complementary. It is thus interesting to start from the high-level view of the Country-level*

graph and go down the levels to analyze in more details specific parts of the routes.

In the present case we decide to make use of the AS-level graph to compare the sequence of traversed ASes before and after the change of origin AS. The origin and destination AS before the change belongs to a backbone ISP, which advertises an aggregated IP prefix including the destination IP prefix. The unreachability of the destination AS after the change can be observed on day four and correlated with the Traceroute Destination Anomaly seen on the same day in the Target History Visualization. Also, the last AS that could be reached by traceroutes appears in the collected BGP AS paths, as the next-hop AS, which is the direct upstream provider, of the new origin AS. This provider-customer relationship could not be officially explained. Hijacking a network can actually be performed by advertising it with a correct origin AS and by putting the attacking AS as the next-hop AS.



▲ **Figure 4.7** — Graph visualization in *VisTracer*. The node-link diagram with embedded clock glyphs shows significant difference in the ASes traversed between the third and fourth day. Reprinted from [84]. © 2012 ACM.

After an investigation, it turned out that the next-hop AS belonged to a company providing DDoS mitigation as service by sink holing the attacking traffic of their customers. The analysis suggests that either the security company redirected the traffic of their customer’s AS because they were under attack or the security company may sometimes act as an ISP for some companies’ AS to easily protect them from undesired traffic. Given the fact that the security company advertised the route in BGP for at least three days, we believe that it actually acted as an ISP for its customer.

Although we have detected abnormal routing changes regarding this network, it is quite difficult to validate these anomalies as a real hijack case since we lack the feedback from the owner of the network.” [84]<sup>7</sup>

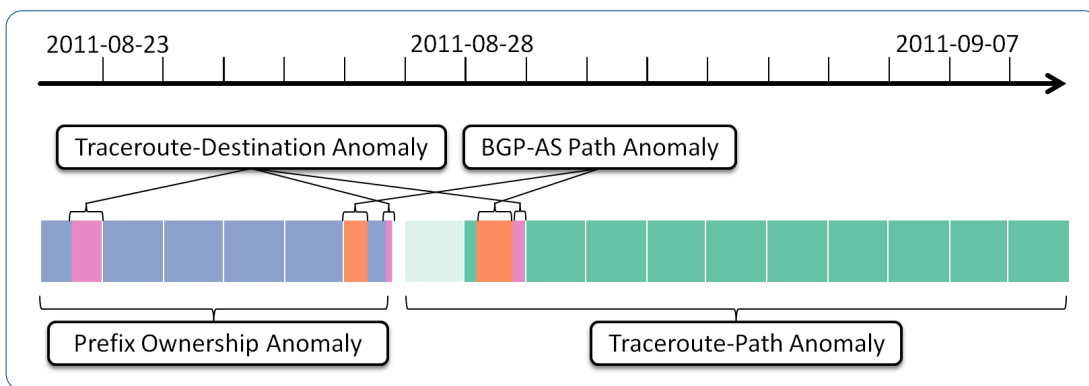
### Case Study: Link Telecom BGP Hijack

This second case study presents the visual analysis of a *validated* BGP hijack performed by a spammer to send spam from the stolen IP address space. The general hijacking

<sup>7</sup> The case study is mostly written by BGP security expert Pierre-Antoine Vervier and is also part of our joint publication [84].

spammer phenomenon has already been observed in [197, 120] and consists of spammers taking control of unused IP address space in order to send spam from clean, non-blacklisted IP addresses.

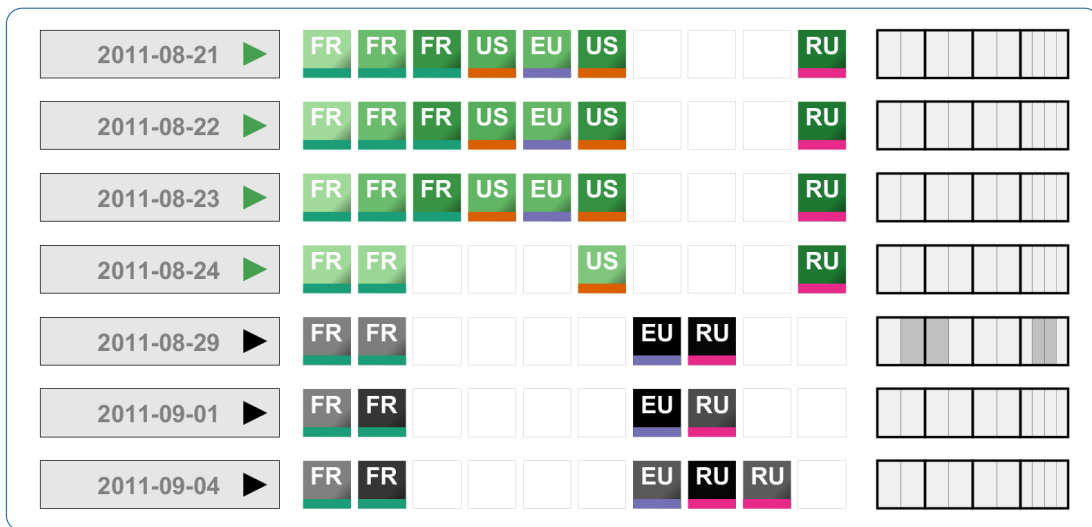
*“From the ASN Overview (Figure 4.8), AS31733 caught our attention, because many diverse routing anomalies occurred within a limited period of time. Moreover, several anomalies occurred on the same day, which reinforced the idea that a major routing change occurred at that time for this AS. The uncovered anomalies related to AS31733 include (i) Traceroute Destination Anomalies (related to the destination host and AS reachability), (ii) Traceroute Path Anomalies and, (iii) BGP AS Path Anomalies (AS Path Deviation).”*



▲ **Figure 4.8** — *Visual ASN Overview of AS31733*. The glyphs reveal many different anomalies over a longer period of time. *Reprinted from [84]. © 2012 ACM.*

*The Target History Visualization of a monitored host within AS31733 exhibiting a combination of Traceroute Destination Anomalies, Traceroute Path Anomalies and BGP AS Path Anomalies. Figure 4.9 presents the Target History Visualization which shows the set of ASes traversed by traceroutes from the vantage point in France to AS31733 throughout the monitoring period. We can clearly see that the set of traversed ASes changes significantly. By looking at the anomalies extracted for that case, we can also see that all anomalies were observed on a particular day, i.e., just after the change in the traceroute path. The observation of the set of IP hosts traversed by the traceroutes shows the exact same behavior. From these observations we can say that the location of the monitored AS in the Internet AS topology changed significantly.*

*Figure 4.10 presents the Graph Visualization of the same monitored host within AS31733. This visualization shows the sequence of IP hops, ASes or countries traversed by the traceroutes. In this case, looking at the Country-level paths would show that packets always seem to go through the US to go from a source in France to a destination in Russia. While this routing behavior can be considered abnormal, we also know that some big ISPs, i.e., backbone ISPs, are spread across continents and may be introduce US hops in a European route. If we now look at the AS-level graph we can*

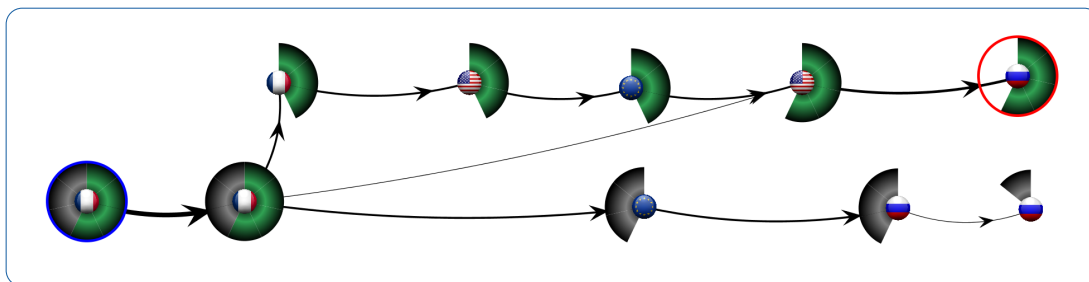


▲ **Figure 4.9** — *Target History Visualization* of multiple traceroutes. The figure shows the significant difference in the *set* of ASes traversed between the fourth and fifth day. The routing anomalies observed are also shown. *Reprinted from [84].* © 2012 ACM.

see that US ISPs Level-3 (AS3356) and Internap (AS12182) both appear in the routes. Besides being a backbone ISP, Level-3 also appears in every traceroute during the monitoring period. However, Internap only appears in the first traceroute, before the routing change. To have more details about the traceroute going through AS12182 Internap, we can have a look at the IP-level graph. The graph reveals that the first traceroute goes through two routers of AS12182 apparently located in the US and then directly ends in AS31733 apparently located in Russia. This suggests that the destination host currently using an IP of AS31733 is likely located in the US instead of Russia. Furthermore, the visualization also shows that the destination host and AS could not be reached from the fifth day until the end of the monitoring period. This observation is corroborated by the Traceroute Destination Anomalies (related to the host/AS reachability) uncovered on the fifth day. All this suggests that the routing change observed lead to the destination host and AS to become unreachable.

After the investigation, it turns out that on August 20<sup>th</sup> 2011 the network administrator of the Russian telecommunication company “Link Telecom”, whose AS31733 belongs to, complained on the North American Network Operators’ Group (NANOG) mailing list that his network had been hijacked by a spammer [222]. On both August 25<sup>th</sup> and August 29<sup>th</sup> 2011 changes were observed in the traceroutes and BGP routes towards AS31733. These changes were the result of the owner regaining control over his network. In this case, the aggregation in the ASN Overview of the routing anomalies extracted for the individual monitored hosts within their AS actually uncovered the pattern of several diverse and timely close routing anomalies.

We further described this hijack case in Symantec’s Internet Security Threat Report 2012 [230]. Although the prefix appeared to be announced



▲ **Figure 4.10** — *Temporal Graph Representation of the confirmed BGP hijack.* The figure shows the significant difference in the *sequence* of ASes traversed. It also highlights the unreachability of the destination AS after the routing change occurred. Reprinted from [84]. © 2012 ACM.

*by the correct origin AS, i.e., AS31733, it was routed via a US ISP called Internap (AS12182). During this period the network was under the control of the spammer, spam messages were received by Symantec.cloud honeypots. The hijack lasted for five months from April 2011 until August 2011 and is a validated case of a hijacking spammer that managed to steal someone else’s IP space and sent spam from it.” [84]<sup>8</sup>*

### 4.2.3 Conclusions and Limitations

In the previous sections we presented a novel visual analytics tool called *VisTracer* to investigate routing anomalies and BGP hijacks. In particular, spamming activities were monitored with the help of a large-scale traceroute collection system. In contrast to related work, our approach was the first method using visual analytics to combine control- with data-plane data sources, to investigate BGP anomalies with the focus on spam campaigns. Special care was taken to design *VisTracer* to support the workflow of analyzing the large-scale dataset according to the analysts’ needs. The tool’s flexibility is derived from the integration of several linked data views and visualizations into a powerful analysis suite, which can address a variety of analysis questions. Furthermore, the usefulness and effectiveness of *VisTracer* for network security analysts was demonstrated in two case studies conducted by BGP security experts. The results and events identified with *Spamtracer* and further explored with *VisTracer* could also be used by our colleagues from Symantec to be incorporated in Symantec’s *2012 Internet Security Threat Report* [230]. However, the actual validation of highly suspicious IP prefix hijackings still remain challenging. Regular usage of *VisTracer* by our analysts would show, which additional views and techniques should be integrated, to even better support the analysis of future BGP and routing-related threats. To improve the scalability of the graph representation, further layout improvements would be beneficial to reduce possible clutter of traceroutes with very complex connections and to incorporate missing hops in the layout.

<sup>8</sup> The case study is mostly written by BGP security expert Pierre-Antoine Vervier and is also part of our joint publication [84].



### 4.3 Visual Analysis for Malware Behavior

This section builds on the following joint publication [261]<sup>9</sup>:

M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner. A Survey of Visualization Systems for Malware Analysis. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*, Italy (Cagliari), 2015. The Eurographics Association. doi:10.2312/eurovisstar.20151114 [261].

*“Malicious code (or malware) is defined as software that fulfills the deliberately harmful intent of an attacker”* [175] and is a major threat in our modern computer networks. Symantec’s *2015 Internet Security Threat Report* [231] confirms again that e-crime and malware is still an increasing major threat. Especially, so-called ransomware attacks have *“more than doubled in 2014, from 4.1 million (...) up to 8.8 million”* [231]. Ransomware is a specific type of malware, either scaring the user with fake warning messages, restricting the access to the computer, or even automatically encrypting the whole file system. Eventually, ransomware demands money from the victim to provide means to get their data back. To pay the ransom, the victim is often requested to pay in bitcoins, which makes it quite difficult to track and shut down such scams [231]. While it is generally not advisable to indulge in such kinds of blackmail, the victims often have no other choice, if they do not have any off-site backup of their data.

A major challenge to fight against malware, is the sheer number of new malware files collected every day. Malware authors make use of sophisticated methods to introduce polymorphism. As a result malware samples differ from each other, so that automatic detection based on simple hash functions is not possible any more. However, anti-malware companies and organizations still need to classify these constantly modified files, which are often obfuscated, to the actual underlying malware family. Therefore, malware analysis often has to focus on the behavior, which is more characteristic for a malware family. In malware analysis, therefore, security experts and researchers focus on *“the process of determining the behavior and purpose of a given malware sample”* [175] that helps to find such common characteristics to eventually define robust signatures to detect malware. This comprises static and dynamic malware analysis. In static analysis, the suspicious file is processed and disassembled to reveal common patterns, so that it can be distinguished from known, or identified as new malware family. In dynamic malware analysis, the malware sample is actually executed within a sandbox environment. Tools observe and log the behavior of all running processes. These behavior logs are then analyzed and compared to known characteristics. Both analysis approaches can benefit from visualizations. Visual analytics can also help to enhance situational awareness especially with respect to the projection stage, because knowing the capabilities of a given malware sample involved in a successful compromising attempt, helps to forecast and assess the consequences.

<sup>9</sup> This published STAR report was eventually the outcome of an initial joint research meeting in Konstanz. All authors were involved in writing and contributed to the paper. Markus Wagner had the overall lead and guided the literature review. Markus and I classified all methods, and I introduced the initial taxonomy draft, which we further refined together. In the paper, I primarily had the lead on Section 5, which I also present in this dissertation. Together, we identified various open gaps and discussed further research directions.

▼ **Table 4.4** — **State-of-the-art overview for visual malware analysis.** Overview of state-of-the-art techniques for visual analysis of malware behavior.

Method	Use Case			Visualization Technique														Visualization	Year												
	Attack Patterns	Routing Anomalies	Malware Behavior	Attack Attribution	Glyph	Node Link	3D Node Link	Scatter Plot	3D Scatter Plot	Color Map	Treemap	Parallel Coordinates	Histogram	Timeline	Tables	Matrix	Geographic Map	Small Multiples	Word Cloud	Pixel Visualization	Radial Visualization	Chord Diagram	Sunburst Chart	Other Standard Charts	Standard 2D Display	Standard 3D Display	Geometrically-transformed Display	Iconic Display	Dense Pixel Display	Stacked Display	
Yoo [278]	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	✓	-	✓	-	2004
Panas [185]	-	-	✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2008
Conti et al. [51]	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	2008
Quist and Liebrock [195]	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2009
Trinius et al. [252]	-	-	✓	-	-	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	✓	2009
Grégio and Santos [105]	-	-	✓	-	✓	✓	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2011
Nataraj et al. [176]	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	2011
Quist and Liebrock [196]	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	2011
Anderson et al. [9]	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	2012
Grégio et al. [106]	-	-	✓	-	✓	✓	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	✓	-	2012
Saxe et al. [206]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	✓	✓	✓	2012
Yee et al. [275]	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	✓	2012
MalwareVis [290]	-	-	✓	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	✓	✓	-	2012
Han et al. [109]	-	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	✓	-	2013
Paturi et al. [188]	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	2013
Wu and Yap [272]	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	2013
Kancherla and Mukkamala [134]	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	✓	-	2013
Donahue et al. [63]	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-	-	-	-	-	✓	-	-	✓	-	2013
SEEM [104]	-	-	✓	-	✓	-	-	✓	-	-	-	✓	-	✓	-	-	✓	-	✓	-	-	-	-	-	-	✓	-	✓	✓	✓	2014
Long et al. [162]	-	-	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	2014
Shaid and Maarof [213]	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	-	-	✓	-	-	2014
DAVAST [267]	-	-	✓	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-	-	2014
Han et al. [111]	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	-	✓	-	-	✓	-	2014
Han et al. [110]	-	-	✓	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	✓	-	2014
Shaid and Maarof [212]	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	✓	-	-	✓	-	2014

In the following section, we will provide an overview of visualization systems for malware analysis, which is an emergent field of research in security visualization, as more and more methods have been proposed in recent years as shown in Table 4.4. More precisely, we provide a taxonomy of existing visualization systems. Furthermore, we identify future research perspectives to eventually enhance malware analysis through visual analytics.

### 4.3.1 Taxonomy of Visualization Systems for Malware Analysis

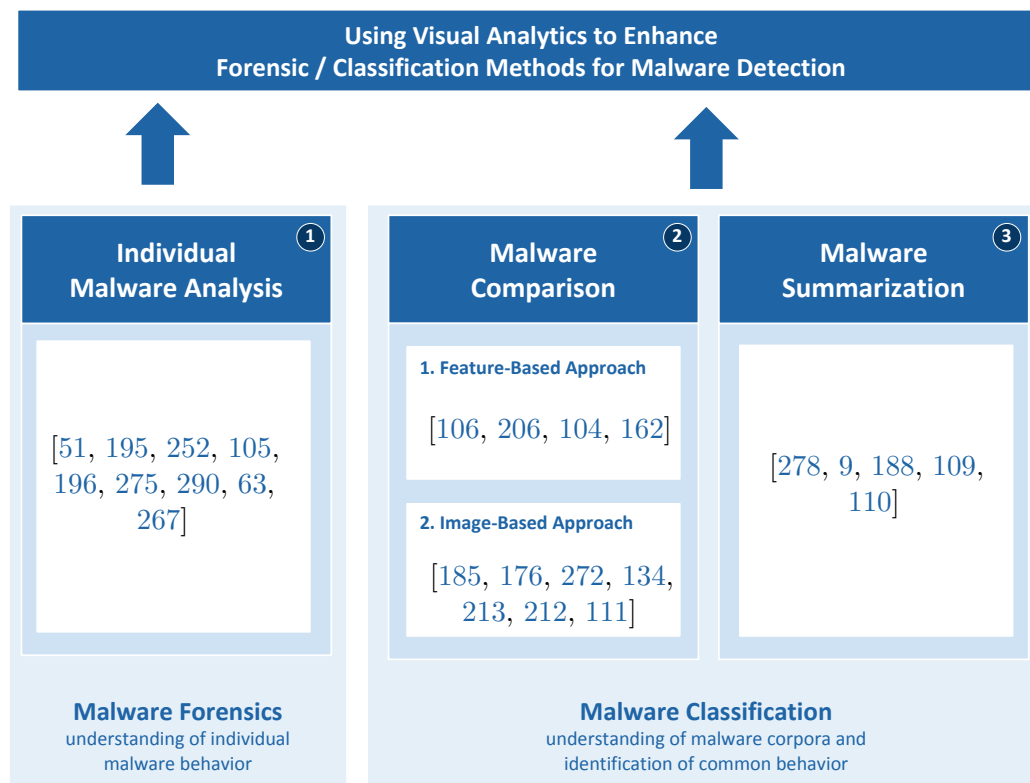
Using visualization obviously helps to understand malware behavior, which is helpful for forensics and *malware detection*. Additionally, visual analysis can help to support the *malware classification* process. Malware detection does mostly refer to the automatic identification of malware (e.g., anti-virus software for end users), however, in more complex scenarios, targeted attacks, or for unknown malware, manual analysis by malware experts is inevitable. Such analyses help to identify suspicious behavior, to

eventually create rules and signatures, which can then be used to improve automated malware detection. Malware classification focuses on the aspect to assign an unknown malware sample to a known group of malware types.

In general, we identified two different main goals of malware visualization systems. On the one hand, there are systems for malware forensics which are used to understand the individual behavior of a malicious malware sample and on the other hand, there are malware classification tools which are used to identify the common behavior of malware samples. Based on these main groups, we differentiate between three underlying main categories and propose a *Malware Visualization Taxonomy* as seen in Figure 4.11. We, therefore, define these categories as follows:

- **Individual Malware Analysis** – These systems support the individual analysis of primarily single malware samples to gain new insights of its individual behavior related to malware forensics.
- **Malware Comparison** – This category fits to visualization tools that are primarily used for comparison of various malware samples for the identification of common behavior (e.g., the malware family) to support malware classification. In general, we have identified two different subcategories:
  - **Feature-Based Approaches** – These systems explore and compare different malware samples based on extracted features and use various data visualization techniques to compare characteristics with each other.
  - **Image-Based Approaches** – Visualization tools in this category generate visual images based on binary data or the behavior logs of the malicious software. Eventually, those visual fingerprints are compared using computer vision techniques.
- **Malware Summarization** – Systems of this category summarize the behaviors of many different malware samples or whole malware corpora to identify similarities and to gain new insights of their common general behavior.

As sketched in Figure 4.11, eventually, one or several malware analysis tools can be used in combination to generate rules and signatures for malware samples or malware families based on the generated insights. Additionally, the increasing use of visual analytics methods will enhance the forensics and classification methods for malware detection. From the taxonomy as seen in Figure 4.11, it becomes obvious that 9 tools focus on individual malware analysis, 11 on malware comparison, and 5 on malware summarization to provide visual summaries of large amounts of malware samples and their characteristics. Additionally, it is interesting to see that only 4 tools for malware comparison are using primarily the feature-based approach, while 7 focus on image-based approaches. Based on the various publication years, it becomes apparent that using malware characteristics (based on features extracted through static and dynamic malware analysis) is becoming more common since 2013 and that fewer systems focus on individual malware analysis (malware forensics). Most of the research for individual malware analysis was performed between 2004 and 2012. In the past 10 years, visualization seems to be used more often to generate image-like representations of malware samples which are then used for visual comparisons.



▲ **Figure 4.11 — A Taxonomy of visual methods for malware analysis.** – Categorization of malware visualization systems into three categories, namely (1) Individual Malware Analysis, (2) Malware Comparison, and (3) Malware Summarization. All systems have the ultimate goal to generate rules and signatures for fully-automated malware detection systems. While the first category tackles the problem of understanding the behavior of an individual malware sample for forensics, the latter two focus on the identification of common behavior for malware classification.

### Visualization Support for Individual Malware Analysis

The first group contains visualization systems geared towards the extensive analysis of *individual* malware samples [51, 195, 252, 105, 196, 275, 290, 63, 267]. Zhuo and Nadjin [290], for example, focus on only one specific type of malware behavior – the network activity of a malware sample – which is then visualized by a glyph-like chart. This specific feature can be explored in great detail which is not possible in other, less specialized visualization tools.

Other tools consider various features at the same time, but still focus on the individual analysis of single malware samples. Trinius et al. [252] use treemaps and so-called thread graphs to visually analyze system calls executed by the selected malware. While basic comparison is also possible with most of the tools in this category (e.g., using multiple instances of the same tool), they do not specifically support bulk analysis.

*Future Research Directions:* The visual analysis of individual malware samples leads the analyst to a better understanding of the specific behavior and can help to judge

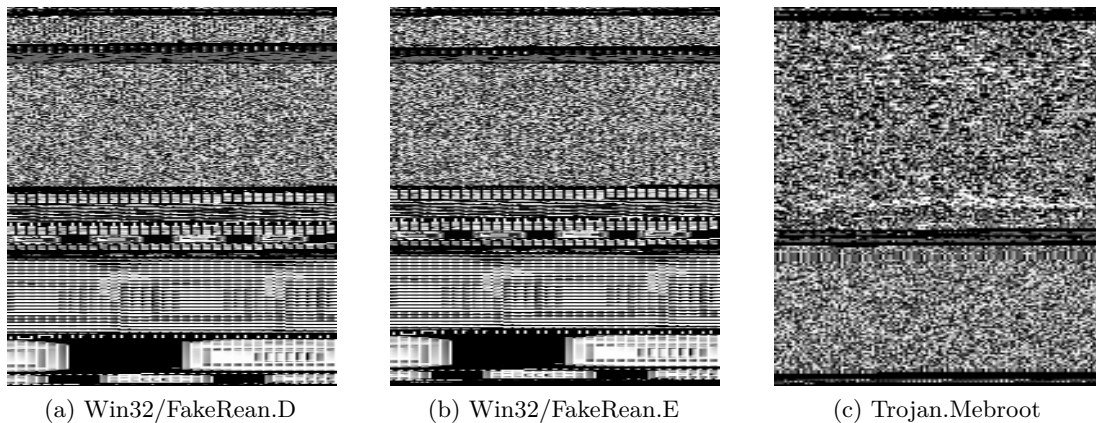
if an unknown sample is indeed malicious or not. However, current work could be improved with respect to malware detection, because many of those tools do not include classification methods to compare the observed behavior to the behavior of known malware types. In the future we expect more visual analytics tools to combine individual malware analysis with automated methods and to incorporate methods to directly relate and compare findings with behavior of known or previously analyzed samples. Automatic highlighting of important or possibly malicious aspects, would help the analyst to quickly focus on most suspicious behavior first to reduce the time needed for manual analysis.

### Visualization Support for Malware Comparison

While the individual analysis is needed to get a deep understanding of a malware sample, the comparison with already known malware samples is crucial for malware classification. On the one hand, this step helps to reduce the number of samples that need time-consuming manual analysis. On the other hand, comparison with other samples can help to identify groups or malware families. All the systems which are represented in this category use visualizations to enhance the comparison of multiple or many malware samples for the identification of their common behavior (e.g., to identify related samples, find the correct malware family). Technically, we distinguish between feature-based and image-based approaches.

**Feature-based approaches** [106, 206, 104, 162] use visual analytics techniques to let the user filter, search, compare, and explore a wide range of properties extracted during analysis. These systems provide means to compare malware samples based on their similarities of features. Individual exploration of these features is also possible, but is much more limited, compared to the previous category. While some of the tools of the previous category were specifically designed to do an in-depth analysis of network activity or to fully explore the temporal sequence of system calls, feature-based malware comparison tools try to focus on a broad set of different features and characteristics, and try to make them all accessible to the analysts. This leads to more abstract representations, higher aggregation levels, and eventually less details for individual features (e.g., ignoring the temporal aspects of network connectivity). The advantage of such approaches is that the analyst can directly compare various features. This helps to understand in which features malware binaries are related and in which they are not.

*Future Research Directions:* The comparison of characteristics helps to visually enhance the malware classification process in various ways. Tools in this category also focus on the question of which features can be extracted and used for comparison. Comparing such malware characteristics helps to identify related samples based on similarity metrics and to identify the common behavior of the explored samples for classification. Especially, the possibility to compare many different features at once and the possibility to apply standard methods from the field of data analysis (e.g., MDS, PCA, clustering) opens a promising research direction. Using visual interfaces to guide the analyst in the selection of features seems to be a good way to better support malware classification. Such visual analytics interfaces would eventually help to define better classifiers to improve malware classification models.



▲ **Figure 4.12 — Comparison of malware images.** Visualizing malware executables as grayscale images is a common technique to visually identify similarities with low computation costs.

**Image-based approaches** [185, 176, 272, 134, 213, 212, 111] have in common that they use visual mappings to render an image for each malware sample. For example, the analyst might need to correlate a given suspicious file to a cluster of malware variants in order to associate the file to a specific malware family. Similar images can be visually clustered using either a manual or an automatic approach based on algorithms from the areas of computer vision and image processing. Some systems visualize the binary data and directly map the (raw) byte-code representation or respective entropy values to an image (e.g., [176, 111]). We applied this technique to variants (D and E) of the *Win32/FakeRean* malware as seen in Figure 4.12 (a) and (b). We use this to detect similar images representing related malware samples. As shown in Figure 4.12 binary images of this particular malware family exhibit quite characteristic and highly similar images, even they represent different files. According to Microsoft threat encyclopedia, *Win32/FakeRean* is a “family of rogue security programs pretend to scan your PC for malware, and often report lots of infections. The program will say you have to pay for it before it can fully clean your PC. However, the program hasn’t really detected any malware at all and isn’t really an antivirus or antimalware scanner. It just looks like one so you’ll send money to the people who made the program. Some of these programs use product names or logos that unlawfully impersonate Microsoft products”<sup>10</sup>.

These particular malware samples can be visually distinguished from Figure 4.12 (c), which represents a *Trojan.Mebroot* malware sample, sharing no visual patterns with the other malware family. This malware “was designed to run undetected on compromised computers and uses a number of sophisticated rootkit techniques to ensure its stealthy execution and thereby prolong the lifespan of the threat. The Trojan modifies the MBR so that it is able to execute even before Windows starts, which means that it is able to bypass security features and create hooks deep in the core of the operating system.”<sup>11</sup>.

Nataraj et al. [177] extract various texture features from such images, to eventually use them for classification. The advantage of this technique is, that it can be applied to any file and can be computed efficiently, which is important for large malware corpora. While classification accuracy is quite comparable for many malware variants,

<sup>10</sup> [microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Win32/FakeRean](https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Win32/FakeRean)

<sup>11</sup> [symantec.com/security\\_response/writeup.jsp?docid=2008-010718-3448-99](https://www.symantec.com/security_response/writeup.jsp?docid=2008-010718-3448-99)

the approach is limited because it does not make use of any dynamic analysis and only relies on the actual bytes found in the binaries. Another problem is, that the visual impression is strongly dominated by possible images embedded in the resource section of an executable, which could be avoided by malware authors to create less obvious visual patterns. To overcome this drawback, the approach was extended to visualize disassembled CPU instructions or API calls (e.g., [185, 212, 213]) in a similar way, however, resulting in higher computation costs.

*Future Research Directions:* One possible future research direction could be the implementation of interaction methods to segment a region of interest or to characterize these texture patterns. Automated image comparison would help analysts to visually identify common code portions or specific instruction blocks within a sample. This information could be used to directly highlight relevant sections in the image. Additionally, the integration and combination of image- and feature-based methods could be promising. Image-based methods using static analysis together with a probability score can be used as efficient first step in a classification pipeline. Afterwards, the more expensive feature-based methods together with dynamic analysis would only be applied to those samples, which share less distinctive image representations, eventually leading to a more scalable classification process.

### Visualization Support for Malware Summarization

While this category is more diverse, the associated tools [278, 9, 188, 109, 110] all provide primarily some kind of summarization capability for a large number of malware samples within the visualization. Some identify a visual mask that is common for all selected samples (e.g., [278]). Others summarize and extract a single combined representative out of many malware variants (e.g., [109, 110]). Finally, some use visual representations to show hierarchical clusters [188] or use heatmaps to visually represent kernels used for a support vector machine classifier to summarize and eventually classify malware samples [9].

*Future Research Directions:* The combination of different types of base data and data provider analysis modes are frequently stated as future work in this category. This will result in larger amounts and more heterogeneous data as input for visualization systems. Another direction into larger amounts of data can be the comparison of malware families as a whole based on their summarization. Finally, the integration of malware summarization with malware comparison and malware forensics using semantic zoom for example is a promising direction.

#### 4.3.2 Conclusions and Limitations

In the previous sections, we provided an extensive overview about the state-of-the-art of visualization systems for malware analysis, which is also presented in an interactive web application<sup>12</sup>. We identified three major categories, namely individual malware analysis, malware comparison, and malware summarization. Each method could be assigned to one of these categories. Future malware visualization systems should investigate more comprehensive designs. For example to switch the perspective between summarization and comparison, or to semantically zoom into individual analysis mode. Likewise the

<sup>12</sup>[malware.dbvis.de](http://malware.dbvis.de)

integration of common features of malware families can be integrated into individual malware forensics to make it more expressive. We also found out, that most systems directly visualize the raw output of the various dynamic or static malware analysis techniques. However, only few systems (e.g., [206]) provide a tighter integration of additional analytical methods to classify or cluster the data. Such integrations and also the stronger visual support in the scope of feature selection and during the actual classification process, seem to be promising future research directions. Enhancing the awareness and a better understanding through visual analytics of common characteristics for misclassified malware samples, would eventually lead to better classifiers and could make the analysis process more efficient.



## 4.4 Visual Exploration for Attack Attribution

The next sections build mostly on the following publication [88]<sup>13</sup>:

F. Fischer, J. Davey, J. Fuchs, O. Thonnard, J. Kohlhammer, and D. A. Keim. A Visual Analytics Field Experiment to Evaluate Alternative Visualizations for Cyber Security Applications. In M. Pohl and J. Roberts, editors, *Proc. EuroVA International Workshop on Visual Analytics*. The Eurographics Association, 2014. ISBN 978-3-905674-68-2. doi:10.2312/eurova.20141144 [88].

The behavioral analysis of attackers in the Internet is a challenging, but highly relevant field of research. It is important to understand their modus operandi to mitigate attacks and develop new methods to protect network infrastructures, customers, and to identify fraud. However, threat actors may belong to various organizations that operate in different ways making it hard to differentiate them based on common behaviour. Fully automated data mining algorithms, and data collecting infrastructures, can help to address this challenge, but when used alone they are often not capable of providing actionable insights, because human analysts can hardly understand the results generated by these algorithms.

Attack attribution is “*primarily concerned with larger scale attacks (...) determining their root causes and (...) deriving their modus operandi*” [57]. Analysts try to relate attacks or malware samples to a larger group or attack campaign. The scope of the following sections is to make use of various visualization techniques to explore and understand inter-related datasets and clusters describing large-scale attack campaigns. The overall goal is to relate new threats to a known group of attackers or campaigns, and help security response analysts to understand the modus operandi and trends within the threat landscape. The three main contributions are: (i) The adaptation of several well-known visualizations to enhance the interactive analysis of threat landscapes. (ii) The integration into *VACS* to visually explore and make sense of the complex results of threat intelligence clustering algorithms. (iii) Sharing results and lessons learned of conducting a field experiment with domain experts.

### Related Work

In the field of visual analytics for cyber security, there are not too many systems, which directly focus on threat landscape analysis to support threat intelligence. In the following, we want to highlight some of the systems, which are directly related to this use case, as also seen in Table 4.5.

Yu et al. [279] propose a system, called *EMBER*, which uses primarily geographical displays to provide a visual high-level overview about extreme malicious behavior. Using normalization based on the population of the various countries, the authors use the map, to highlight regions with unexpected high malicious activity. Their system “*uses a metric called Standardized Incidence Rate (SIR) that is the number of hosts exhibiting*

<sup>13</sup>The responsibilities for this joint publication were divided as follows: I lead the paper and did most of the writing. Together with James Davey and Olivier Thonnard, we conducted the actual field experiment in Dublin. Olivier applied his TRIAGE algorithm, and described the technique. James helped primarily in summarizing the results, while all authors gave feedback, and helped with proofreading.

▼ **Table 4.5 — Related work for analyzing the threat landscape.** Overview of related work to analyze the threat landscape and provide visual support for attack attribution.

Method	Use Case	Data Source	Visualization	Year
	Attack Patterns Routing Anomalies Malware Behavior Attack Attribution	Packet Traces Network Flows IDS/IPS Alerts Firewall Logs Vulnerability Scans Meta Data System Metrics / Status Reports DNS Logs BGP Messages Server Logs File System Changes Audit Trails Webserver Logs Database Logs Honeypot Logs Spam / Phishing Mails Malware Files Behavior Logs	Standard 2D Display Standard 3D Display Geometrically-transformed Display Iconic Display Dense Pixel Display Stacked Display	
EMBER [279]	-	-	-	2010
Nicter [73]	-	-	-	2011
BURN [203]	-	-	-	2011
Tsigkas et al. [253]	-	-	-	2012

*malicious behavior per 100,000 available hosts*” [279]. Such systems help to get a high-level overview of the current threat landscape. While EMBER uses a geographical approach to analyze source IP addresses from detected attacks, collected by honeypots (and other systems) around the world, Roveta et al. [203] focus on rogue autonomous systems (AS). They propose a visualization system, called *BURN* [203] to display and explore temporal data about rogue autonomous systems. The data is also collected via honeypots using the *FIRE* [227] system, which gathers data, for example using honeypots, to “*identify and expose organizations and ISPs that demonstrate persistent, malicious behavior*” [227]. This data is visualized in *BURN* to give an overview about the most malicious computer networks around the world. To make this data accessible, they use various visual representations, and mainly use animated interactive bubble charts to provide a global view. This system, therefore, helps to identify ISPs, which host many malicious websites or illegal services.

While the previous systems, focus on the overall threat activity around the world, Tsigkas et al. [253] focus more on the analysis of threat activity related to specific types of attacks. Tsigkas et al. [253] propose a technique to create abstracted node-link diagrams, to analyze common feature values of spam campaigns to gain “*insights into the strategic behavior of spam botnets and spammers operations*” [253]. The input data is gathered by spamtraps distributed around the world, to gather spam and phishing e-mails. These “honeypots” attracting malicious e-mails, provide a rich dataset to get an in-depth view about ongoing spam campaigns and the behavior of these attackers on a larger scale.

In the following, we propose a set of alternative visualization approaches, tackling the very same problem. In contrast to Tsigkas et al. [253], we not only employ node-link diagrams, but also apply a graph-based clustering algorithm, to analyze the resulting clusters with space-filling techniques and use glyph-based representations in a small multiple setting to provide an overview and compare multi-dimensional clusters, which relate to distinctive malicious campaigns (e.g., spam campaigns).

#### 4.4.1 Data Analytics for Threat Intelligence

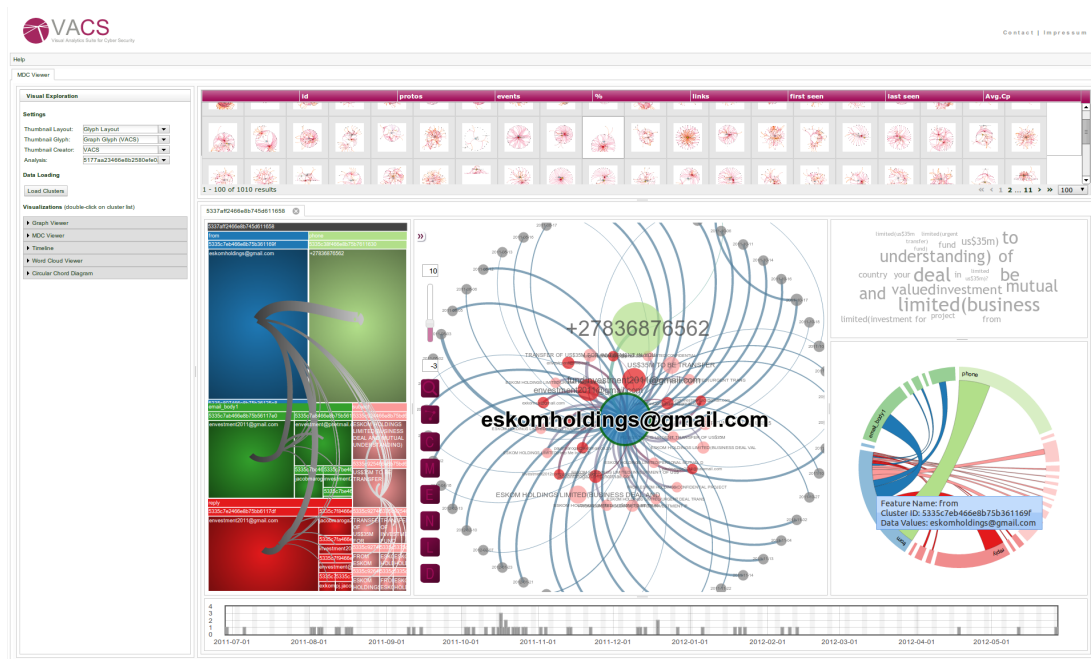
In the following we leverage the multi-criteria clustering algorithm *TRIAGE* [249], which was designed to support threat intelligence and attack investigation tasks. The applicability of this algorithmic approach has been previously shown and evaluated in different uses cases on various datasets [126, 250, 248, 56].

The *TRIAGE* algorithm uses a combination of graph-based analysis and data aggregation methods as used in multi-criteria decision analysis [249]. The system can generally be applied to various security-related datasets consisting of individual events, for the purpose of identifying groups of related events that might have a common root cause, *e.g.*, series of cyber attacks sourced by the same attackers or *threat group*. In a spam e-mail dataset, for example, each message represents one event with different features (*e.g.*, sender address, recipient, subject), denoted as  $F_k$ , with  $k = (1, \dots, n)$ . For each feature an undirected edge-weighted graph  $G_k(V_k, E_k, w_k)$  is created, where the vertices  $V_k$  represent the message features, and the edges  $E_k$  weighted by the function  $w_k$  reflect similarities among messages [248]. Afterwards the different weighted graphs  $G_k$  are combined using an aggregation function. The resulting multi-dimensional clusters (MDCs) represent groups of events correlated by a number of features, where the combination of correlated features may vary within the same cluster, depending on the data fusion model. In a spam dataset such MDCs are likely to reflect individual spam campaigns containing messages having similar characteristics, and hence a common root cause.

#### 4.4.2 Integrated Visualizations for MDC Exploration

To conduct interactive exploration of multi-dimensional clusters, we integrate various additional visualizations into *VACS*, which were afterwards also integrated into Symantec’s visual analytics application containing visual dashboards, charts, tables for feature selection, and cluster visualizations to cover the whole analysis workflow. In the following, we focus on the usage of three visualization techniques as seen in Figure 4.13 that can be used to explore individual MDCs: The *Treemap View (TV)*, the *Graph View (GV)*, and the *Chord View (CV)*.

- **Treemap View (TV)** – This space-filling view as seen in Figure 4.13 provides an overview of the features, mapped to color, and their value occurrences. Each colored rectangle on the upper level represents a feature, containing further rectangles representing cluster prototypes. The more frequently a value, the bigger the corresponding rectangle in the squarified treemap [32]. Interaction enables the user to zoom in and reveal splines to show the event co-occurrences of values in entities. Treemap representations with splines are also used in related security applications [81] for the exploration of network traffic, while treemaps alone are commonly used to provide overviews for forensics [118] and malware analysis [252].
- **Graph View (GV)** – This interactive node-link diagram shows the relationships between feature values, which is widely used in various security applications [170]. Each node represents a value occurring in the cluster, whereas an edge indicates the co-occurrence of a pair of values in an event of the dataset. The node sizes are mapped to the number of events and the thickness of the edges is determined by the number of co-occurrences. The graph is highly interactive and provides, zooming, panning, re-positioning nodes, and the modification of edge thickness,



▲ **Figure 4.13** — Using *VACS* for visual attack attribution. After feature selection and analysis, the shown visualization display can be used to explore the MDC clusters. The small-multiple view at the top can be used to select MDCs. The *Treemap View (TV)*, the *Graph View (GV)*, and the *Chord View (CV)* show the respective MDC of a well-known scam campaign impersonating the company “Eskom Holdings” [126].

label size, and node size. To handle large datasets, a sampling can be applied and the layout is calculated on the server-side using *Graphviz* [65].

- **Chord View (CV)** – This interactive circular chart enables the exploration of all relations between the different feature clusters composing the MDC. The circle segments on the edge of the view represent the values, their colour is determined by their feature. Interactive highlighting shows which feature clusters have co-occurring events. When the users selects a feature cluster, the shown chords encode the number of co-occurring events to the other feature clusters. The implementation is based on *D3.js* [29] using an approach similar to *Circos* [146], which is widely used to analyze complex datasets.

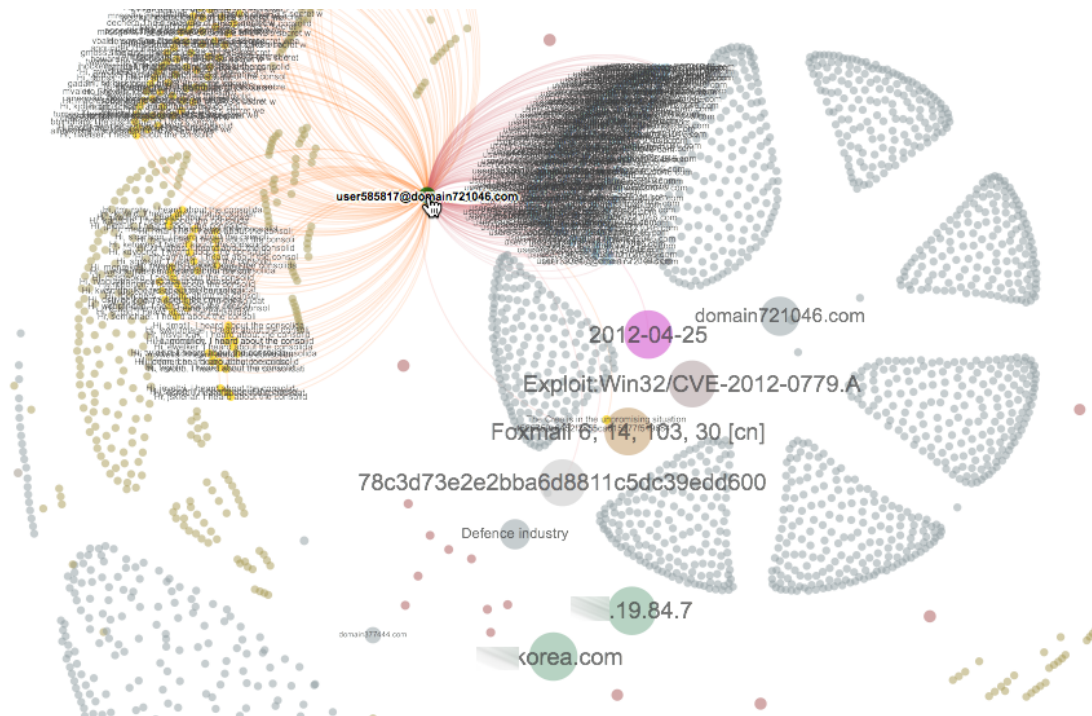
The visualization modules were built on top of *VACS* (as introduced in Section 3.1.1), which is a web-based research framework providing visualizations and a secure REST interface to remote datasets and algorithms of multiple *VIS-SENSE* project partners. This modular architecture helps to interdisciplinarily develop visual analytics applications that enables us to work on sensitive datasets and novel algorithms, while preserving the rights of the property owners.

#### 4.4.3 Evaluation using Field Experiment

In the field of visual analytics, evaluation is quite challenging [255]. On the one hand, real-world scenarios often have no ground truth, and on the other hand, only experts can identify and validate insights. User studies in the lab are not an appropriate

or realistic approach to judge the usefulness of visual analytics applications, which require in-depth domain knowledge. Shneiderman and Plaisant [219] propose the use of “*Multi-dimensional In-depth Long-term Case studies (MILCs)*” [219], which is a promising long-term evaluation approach. However, this is hard to achieve in practice due to the lack of financial support and willingness of experts to participate in such studies. Shiravi et al. [216] states that “*one of the reasons that security visualization systems, despite their great potential, are not often incorporated (...) is the result of failing to address the focal points of user experience*” [216]. We tried to address this issue and gathered feedback about the user experience for the three visualization techniques. Within *VIS-SENSE*, we had the chance to conduct a two-day *field experiment* [38] with security experts from an operational response team while observing them, how they worked with their own data using our visual analytics application deployed as prototype system on their premises.

### Field Experiment with Security Response Experts



▲ **Figure 4.14** — Investigation during the field experiment. Example of an MDC found during the field study, attributed to a notable espionage campaign. *Reprinted from [88]. © 2014 The Eurographics Association.*

The two-day field experiment was conducted in November 2013 and carried out on the premises of Symantec Security Response in Dublin, Ireland and involved six participants with a solid background of cyber security threat analysis. The study was focused on collecting a qualitative assessment of the visual analytics system and to evaluate the user experience of the interactive visualizations. It consisted of three phases. First, a general introduction to the goals and results of the *VIS-SENSE* project were given, followed by an interactive demonstration using data known by the project partners.

Then, the main part of the field experiment consisted in a hands-on session, in which the participants used the system for analyzing their own data. The demonstration was designed to show how a typical exploratory session would be carried out. It illustrated the use of the main functionalities, such as the overview, search, and visualization features and showed how to find, confirm, and explain interesting patterns. The main task was to “*explore clusters to understand the reasons why these entities have been grouped together*” including questions like: Which customers are targeted? What are the strongest correlative features and characteristics of a campaign? What are the most significant coalitions of features that are linking entities?

### Hands-On Session

In the hands-on session, the participants had approximately two hours to analyze their data with our visual analytics system. There were four users (three analysts and one designer) actively using the application on three laptops. Their dataset consisted of 44 features and approximately 100,000 entities and had not previously been analyzed with *TRIAGE* and was completely unknown to us. Due to a lack of experience with the dataset, the parameters for the clustering algorithms were chosen based on what had worked well previously with other similar datasets. Based on density and cardinality 10 features were selected as input to the clustering algorithm. In spite of these challenges, we were able to find clusters suitable for exploratory analysis and hypothesis formulation and validation. An example is illustrated in Figure 4.14, which represents a notable cyber espionage campaign that affected two large defense industries in April 2012, and was attributed to the Elderwood gang [229]. To acquaint the analysts with the software, a series of simple, predefined interactive tasks, and general questions were given to the participants. However, the analysts were able to freely use the software to explore their dataset. We observed the participants passively, but were available on request to answer questions and provide guidance to the participants. After the hands-on session an informal discussion was conducted and feedback of the participants was recorded. At the end of the experiment, a summary was presented to all participants and further interested parties.

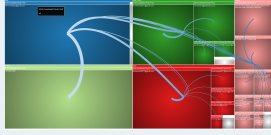
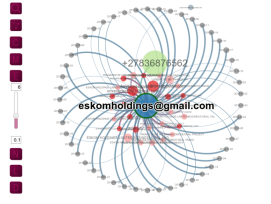
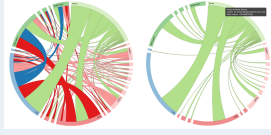
### Results of the Field Study

During the introductory presentations the participants posed detailed questions about the techniques, hardware, and software. They appeared to see the applicability of the visualizations to their own work. In particular, the participants wanted to know more about the potential for the integration of visualization components into other environments. They stated that their goal in the field experiment was to try out the components to see whether they could achieve the tasks, they previously had to do manually, faster with our technologies. We found this encouraging and it explained the overall high degree of interaction during the meeting. The users began working on the tasks set for them during the hands-on session and had little trouble achieving basic tasks. In some instances, a few words or a sentence from the observers was required to guide the participants, but no deeper explanations were necessary. The participants diverged from the structured tasks frequently to engage in more exploratory activities, returning occasionally to the tasks. In this way they were able to *test out* each of the interactive features of the interface. Some participants started targeted searches for specific phenomena in the data, copied attributes from the application and

compared them manually with other datasets and internal systems. One participant began a deeper exploration of a cluster in GV, repositioning nodes and conducting a closer examination of connections. The cluster showed a cyber-criminal campaign. The participant was able to identify and characterize distinct phases of the campaign and used the visualization to explain the modus operandi of the attacker to another participant. Other similar spontaneous discussions between users about their findings occurred, which showed that our system fostered collaborative analysis.

The participants had some difficulty acquainting themselves with the UI due to missing UI features and lack of UI documentation. For example, they applied filters in tables and expected similar filtered views of the data in the visualizations. However, this feature was not yet implemented and the lack of linking in the displayed data led to some confusion. They also complained about the lack of meta-data integration. In general, GV was perceived as the most useful of the three alternative visualizations. Indeed, GV was used most intensively by the participants. To avoid the influence of layout and positional preference and to force the analyst to focus on each visualization individually, each visualization was shown in full-screen and not as integrated display as seen in Figure 4.13. The other two were tried out initially, but not pursued much subsequently. Participants generally preferred TV to CV while the latter was criticized as lacking usefulness for their workflow. However, it still may be useful for short overviews of relations in very large datasets. A participant commented that their most common workflow is of an investigative nature; drilling down into the data and exploring details. Thus, visualizations focused on providing an overview without possibilities for deeper interactive exploration are not very useful for them. In addition, GV was the most interactive of the three views. Overall, it was concluded, that GV was best suited for their need of detailed structural exploration for medium sized MDCs, TV provided an helpful and compact overview, while the least preferred CV mostly focused on exploration of relations between clusters within a MDC. A high-level overview summarizing the qualitative feedback is presented in Table 4.6. A participant commented that the system did open many new possibilities for data exploration and representation. The system was perceived as very useful to speed up analysis tasks. Furthermore, the participants provided many constructive suggestions for improvement, in particular for further enhancing user interactions and data analytics capabilities.

▼ **Table 4.6 — High-level summary of qualitative feedback.** A selection of results based on qualitative feedback during the field experiment.

Visualization	Best Task	Usefulness	Pros & Cons	
	<b>Treemap View (TV)</b>	compact overview	useful	<ul style="list-style-type: none"> <li>+ compact overviews</li> <li>+ good scalability</li> <li>– splines not lockable</li> </ul>
	<b>Graph View (GV)</b>	detailed structural exploration	very useful	<ul style="list-style-type: none"> <li>+ high interactivity</li> <li>+ encouraged discussion</li> <li>+ investigative workflow</li> <li>+ most time spent</li> </ul>
	<b>Chord View (CV)</b>	cluster relations	limited use	<ul style="list-style-type: none"> <li>– fixed layout</li> <li>– hovering not lockable</li> <li>– navigation issues</li> </ul>

#### 4.4.4 Conclusions and Limitations

We presented a web-based visual analytics application to analyze multi-dimensional clusters to support the *TRIAGE* algorithm and enhance attack investigation tasks associated with it. We conducted a field experiment to gather qualitative feedback from domain experts specifically on the usage of three visualization techniques. Furthermore, we identified primary tasks for these alternative visual representations on the basis of the feedback of the analysts. The detailed feedback can be summarized in three areas, which can guide future research directions:

- Parametrization for the clustering of unknown datasets proved to be challenging, which further strengthened the importance of visual feature and parameter selection to make justified decisions.
- The feedback showed the importance of highly interactive visualizations. Slight improvements (e.g., filtering, highlighting multiple elements) in the visualizations can lead to considerable changes in user experience and it may have a strong impact on the usability to solve real-world problems.
- Inconsistent design decisions easily cause confusion. In collaboratively developed software, inconsistencies in design are common but should be avoided.

The research prototypes have since been integrated into Symantec’s internal research framework to analyze security datasets and are actively used for other activities.



## 4.5 Conclusions

In this chapter, we investigated some critical use cases of visual analytics with respect to network threats. Identification and understanding the behavior of cyber security threats is an important factor for situational awareness. In the first section, we made use of a generic visualization technique, called *TMDS*, to be used in the context of attack pattern visualizations, and evaluated its applicability. Furthermore, we addressed one of the most critical threats attacking the fundamental routing in the Internet, which are BGP prefix hijackings. To provide means to support analysts, investigating such incidents, we developed a visual analytics system, called *VisTracer*, to explore occurred anomalies with respect to actual spam campaigns. Because of the recent trend in visualization systems for malware analysis, we conducted an extensive literature review, and provided a taxonomy for visualization methods in the context of malware analysis. Additionally, we discussed future research directions, which suggest further usage of visual analytics in this highly relevant field. Eventually, we focused on attack attribution and the overall threat landscape, in which visual analytics can be used to support security response teams in their daily work analyzing ongoing threats on a larger scale. This is done by integrating various visualizations to support the visual exploration of multi-dimensional clusters into *VACS*. To evaluate this approach, we conducted a field experiment in the premises of an operational security response team. The usage of our visual analytics systems on their own data was very well received and they provided extensive feedback which will be valuable for future research in visualization support for threat intelligence.



*“It is just good practice to do a situational scan and have situational awareness when you are out in the world.”*

— Phillip C. McGraw



## Visual Analytics for Network Streams

### Contents

<b>5.1</b>	<b>Visual Overview for Stream Monitoring</b> . . . . .	<b>148</b>
5.1.1	Usage of Dynamic Visualizations for Stream Monitoring . . . . .	149
5.1.2	Conclusions and Limitations . . . . .	149
<b>5.2</b>	<b>Visual Correlation for Heterogeneous Data Streams</b> . . . . .	<b>150</b>
5.2.1	NStreamAware – Scalable Analytics for Data Streams . . . . .	150
5.2.2	Conclusions and Limitations . . . . .	152
<b>5.3</b>	<b>Visual Exploration for Sliding Windows</b> . . . . .	<b>153</b>
5.3.1	NVisAware – Visualization Technique for Sliding Slices . . . . .	153
5.3.2	Evaluation using Network Security Case Study . . . . .	157
5.3.3	Evaluation using VAST Challenge 2014 . . . . .	160
<b>5.4</b>	<b>Limitations and Conclusions</b> . . . . .	<b>163</b>

GAINING situational awareness in the field of cyber security, when real-time network streams are involved, is often not a static process, but a *dynamic visual analytics* [167] problem. To gain insights a tight coupling of *interactive monitoring* and *visual exploration* is needed. However, the previous chapters, and also most of the state-of-the-art do not explicitly focus on the visualization challenges for incrementally changing heterogeneous real-time data. This is also stated by Shiravi et al. [216], that security “*visualization systems, in their current state, are mostly suitable for offline forensics analysis.*” [216], while real-time “*processing of network events requires extensive resources, both in terms of the computation power required to process an event, as well as the amount of memory needed to store the aggregated statistics*” [216].

To emphasize the importance of visual analytics systems, explicitly supporting the analysis of evolving data streams in a real-time 24/7 monitoring fashion, we completely devote this chapter to such a challenge. Furthermore, we show that our approach is quite generic for heterogeneous data streams, which makes it relevant for more than one of the security-related use cases described in the previous chapters.

## Situational Awareness for Data Streams

This chapter builds mostly on the following publications [79, 80]<sup>1</sup>:

F. Fischer and D. A. Keim. NStreamAware: Real-Time Visual Analytics for Data Streams to Enhance Situational Awareness. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 65–72, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671495 [79].

F. Fischer and F. Stoffel. NStreamAware: Real-Time Visual Analytics for Data Streams (VAST Challenge 2014 MC3). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 373–374, 2014. doi:10.1109/VAST.2014.7042572 [80].

In many security-related scenarios the analysis and situational assessment of data streams is crucial to detect suspicious behavior, to monitor and understand ongoing activities, or to reduce streams to focus on the most relevant parts. As discussed in Chapter 3 there are various data sources for network activity monitoring. Network routers and servers, for example, produce a continuous stream of network flow records or system log messages, and hundreds of system metrics and performance data. In some times, analysts do a close real-time monitoring, while in other situations analysts have no choice, but to focus only on the most important parts of a data stream. The same is true in the field of law enforcement in the analysis of criminal activities of ongoing threats to maintain situational awareness. In this scenario, analysts need to handle streams of possibly important social media messages and call center messages. Both scenarios are technically related and show the high importance of research in the field of data stream analysis with the analyst in the loop that is a key to enhance situational awareness. The challenge in this field is also to merge and aggregate heterogeneous high velocity data streams. While we do have a wide variety of highly-scalable databases and there has been much research in intrusion and anomaly detection, fully automated systems are not working sufficiently. To convey and support understanding, generate insights, and evaluate hypothesis, analysts need to have a central role in such a system, to not lose context, and to be able to judge data provenance. The ultimate goal allows the analysts to actually get an idea what is going on in a data stream to gain situational awareness. Such analysts are often “*being asked to make decisions on ill-defined problems. These problems may contain uncertain or incomplete data, and are often complex to piece together. Consequently, decision makers rely heavily on intuition, knowledge and experience*” [200], which highlights the need to guide analysts to the *right* parts of a data stream, because it is impossible to explore everything in the same level of detail.

In the following sections, we introduce *NStreamAware*, which is a visual analytics system designed to address this challenge using latest technologies available from the big data analysis community [280] and real-time visual analytics research [167].

The main contributions of this chapter are the following: Firstly, a system architecture, called *NStreamAware*, based on *Apache Spark Streaming* [11] to summarize

<sup>1</sup> The writing, implementation, and programming was done by myself and successfully published at VizSec [79]. Daniel Keim gave advice and suggestions on the project. The application to the VAST Challenge dataset was also done by myself with some support of Florian Stoffel for the challenge submission [80].

incoming data streams in *sliding slices*. Secondly, a web-based visual analytics application, called *NVisAware*, using a novel combination of various visualization techniques within multiple sliding slices to visually summarize the data stream based on selected features steered by a visual analytics interface.

### Design Considerations

Based on the given problem, experience, and expert feedback with earlier work in the field, we identified following design considerations and principles as crucial for our approach.

- DC1 Incorporation of novel scalable analytics methods** – Scalable, distributed, and proven large-scale analysis frameworks must be building blocks of a system able to address big data problems. We need to take advantage of such novel technologies from the big data community and use them in visual analytics application. We need to bring those worlds together and keep the analyst in the loop to address complex problems.
- DC2 Enabling real-time monitoring** – While it is not possible to present *all* raw messages for high speed streams, it is still relevant for many scenarios, where analysts want to closely monitor messages from a particular system, or based on a specific filter criteria in real-time. Many available visual analytics systems, however, still do require a static batch loading first. We see the need to be able to directly push data to our system in a streaming fashion, and be able to smoothly switch between monitoring and exploration.
- DC3 Deterministic screen updates, independent from data stream** – The problem in systems supporting DC2 is the high cognitive load for the analysts when analyzing real-time streams. Because of the unpredictable characteristics of data streams with respect to volume, velocity, variety, and veracity, we additionally need visualizations able to decouple the flow-rate of a data stream from screen updates and keep the latter constant and predictable to not overwhelm the user. There is a trade off between DC2 and DC3 to achieve both at the same time.
- DC4 Fusion of heterogeneous data sources** – Many available systems do focus on individual data sources, and provide less flexibility to incorporate and correlate various heterogeneous data sources. However, focusing on particular individual data sources helps to develop highly effective specific visualization systems. On the other hand, it is important to cover a broader field of scenarios and tasks, to provide better situational assessment.
- DC5 User-steered feature selection** – Feature selection is an important field to support analysts using appropriate visualization and interaction techniques. Our goal is to enhance understanding of data streams and provide more compact overviews. In this process, we want to integrate the human in the workflow that requires a tight coupling of visual representations, interaction and analytic methods.

### Related Work

The contributions of this work are related to various research fields, so we discuss various areas in the following section. Many researchers focus on the algorithmic analysis of data

streams, especially in the field of stream clustering [4] and event detection. In recent years, there was a focus on social data streams, because of the wide availability of such data. While most of these systems focus on the detection of events, our work contributes more in the field of visualizing a condensed heterogeneous data stream to focus on more interesting changes, omitting or merging less interesting ranges to eventually focus on important parts in more detail. This idea is related to the work of Xie et al. [274] proposing a fully-automated merging algorithm for *time-series* data streams. A recent study by Wanner et al. [265] takes a look at the evolution of visual analytics applications for event detection for text streams and concludes that “*visualizations were primarily used as presentation, but had no interaction possible to steer the underlying data processing algorithm*” [265]. This confirms our assumption, that many systems do not cover DC5 appropriately. Our approach differs, that we provide interactions, so that users are able to steer the feature selection process. Therefore, the system does not only rely on the fully-automated selection of interesting parts, but on the user-adjusted feature set. The ultimate goal of visual analytics systems for data streams is to enhance situational awareness to facilitate decision making. Endsley [67] provides a widely used generic definition of SA as also described in Chapter 2. It “*is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future*” [67]. Further work makes it clear, that situation awareness primarily resides “*in the minds of humans*”, while situation assessment better describes the “*process or set of processes*” leading to the state of SA [235]. In the complex field of computer network security operations, only a combination of various tools used by experienced domain experts, will eventually be able to guide the user to such cognitive state.

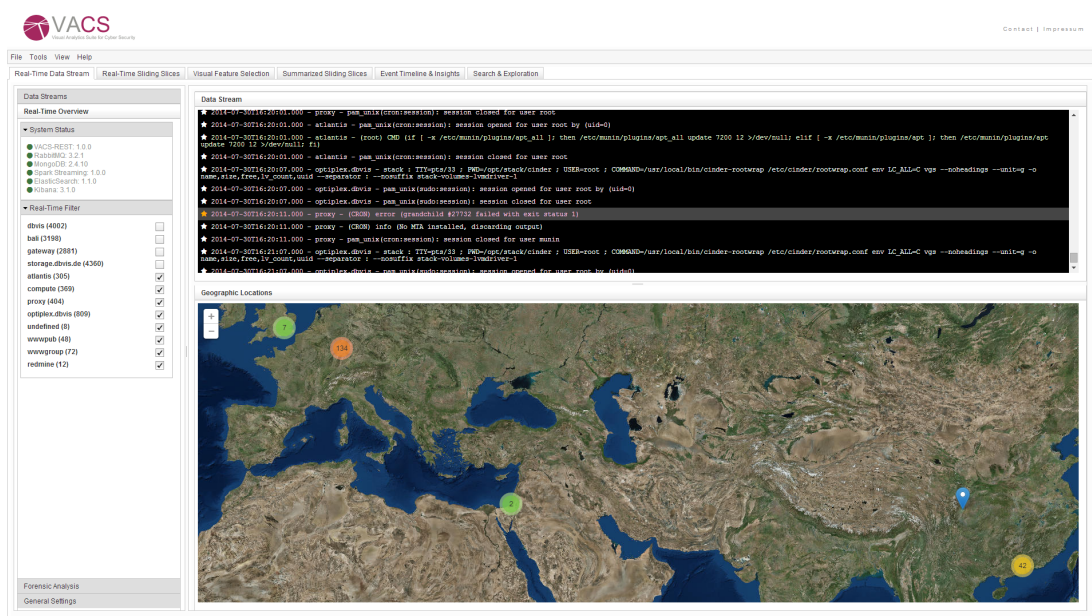
As described in the previous chapters, there is lot of research to enhance situational awareness for network activity and also for network threats using various visualization techniques. Erbacher [71], for example, designed various visualization techniques to explicitly convey the current state of the network to best support situational assessment. *ELVIS* by Humphries et al. [121] is a highly interactive system to analyze system log data, but cannot be applied to real-time streams. *SnortView* [142] focuses on the specific analysis of intrusion detection alerts and also does not satisfy DC2. The focus of previous work of our own, the *Event Visualizer* [85], is to provide real-time visualizations for *event* data streams (e.g., system log data) to provide real-time monitoring and possibilities to smoothly switch to exploration mode covering DC2 and DC4. In contrast to this event-based approach, Best et al. [24] proposes another real-time system to enhance situational awareness using the analysis of network traffic based on *LiveRAC* [172]. The analyzed and aggregated time-series are displayed in a zoomable tabular interface to provide the analyst an interactive exploration interface for time-series data, while the approach, we are going to present, is more general to include also other data types (e.g., frequent words or users, hierarchical overviews) addressing DC4.

## 5.1 Visual Overview for Stream Monitoring

Implementing a visual analytics application, which is capable of providing a basic overview about the raw messages of heterogeneous data streams is initially not very complex. Therefore, we integrated a straight-forward implementation of an interactive display into the *VACS* system (introduced in Chapter 3) to provide an initial real-time display for data streams.

### 5.1.1 Usage of Dynamic Visualizations for Stream Monitoring

Figure 5.1 shows the main display of our web application and includes various filter capabilities to restrict the incoming messages in real-time. In the upper part, the whole incoming textual data stream is shown. This is done similarly to the popular `tail -f` command on Unix-based systems. Various coloring schemes can be applied to highlight different features. This can, for example, be used to visually emphasize uncommon messages, which are not seen frequently. The incoming data stream can also be preprocessed to extract geographic locations (e.g., from known GPS data, or derived from geographic IP or ASN lookups). The map in Figure 5.1 will plot each message to the respective position and applies clustering to improve visual overview.



▲ **Figure 5.1 — Real-time visualization display.** A basic visual display to monitor incoming live streams as raw messages and plot extracted geographic locations to a map. *Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.*

### 5.1.2 Conclusions and Limitations

In this section, we successfully integrated a real-time visualization into VACS to track and filter active and dynamic data streams. However, the main problem of such real-time displays, that represent the raw data without any form of aggregation, is that the screen updates are not independent from the velocity of the data stream. Analytics, ranking, scoring, and filtering can help to focus on parts of the data stream. Real-time monitoring of parts of a data stream is indeed a common use cases. However, this implies that the analyst actually knows, what he is looking for. If he knows so, for example, if he want to focus on all messages of a specific server, or wants to investigate the occurrences of a specific known error messages, such real-time visualization displays

are definitely helpful. However, for large-scale data streams, this is not feasible. Because of the ever changing display, the cognitive load gets too high for a single analyst to make sense out of the data to stay aware of the current situation. So the main question, we need to tackle is: how can we build a scalable system and a visualization, which reduces the cognitive load of the analyst?

## 5.2 Visual Correlation for Heterogeneous Data Streams

Before we are able to address the visualization challenge, to reduce the cognitive load of an analyst, we need to step back and focus on the scalable infrastructure needed to provide means to buffer, correlate, preprocess, and analyze heterogeneous data streams. In the following, we describe the building blocks of *NStreamAware*, which provides scalable analytics for the correlation of heterogeneous data streams.

### 5.2.1 NStreamAware – Scalable Analytics for Data Streams

To process the data stream, we made use of various modern technologies to provide a scalable infrastructure for our modular visual analytics system. Our architecture consists of our *REST Service*, *Spark Service* and a web application integrated to *VACS* with various visualizations. This is the basis to analyze *heterogeneous* data streams, control data fusion, and eventually conduct a visual analysis as seen in the previous section in Figure 5.1. To provide proven and scalable data processing, we make use of *Apache Spark*<sup>2</sup>, *RabbitMQ*<sup>3</sup>, *ElasticSearch*<sup>4</sup>, and *MongoDB*<sup>5</sup>. The overall architecture can be seen in Figure 5.2. The *REST Service* (1) connects to the data streams (2), preprocesses the data, and calculates various additional information for the incoming events. The service does also provide a REST interface to retrieve historical data or manage insights. All events are stored to a distributed *ElasticSearch* cluster and are forwarded to our message broker *RabbitMQ*. The *Spark Service* (3), which runs on top of the *Apache Spark Streaming* [11] platform for analytics, generates real-time summaries on sliding windows, and stores them to a *MongoDB* database (4). *Spark Streaming* is a development framework to help to implement analytical algorithms executed in large distributed cluster environments to provide scalability even in big data scenarios. The *Spark Service* is implemented using Scala and calculates various statistics and features based on sliding windows. The used window size, overlap, and other parameters need to be manually defined to roughly match the data characteristics and analysis goals. Table 5.1 shows a selection of calculated example features for a network security use case. We call these summaries, which are generated in a regular interval, *sliding slices*. Those slices and also a selection of raw messages are eventually forwarded to our web application *NVisAware* (5), so that they can be visualized in the graphical user interface to the analyst using various interactive real-time displays.

All modules are loosely coupled, so that they can be deployed on separate computers or in cluster environments to achieve best performance for large-scale data streams.

---

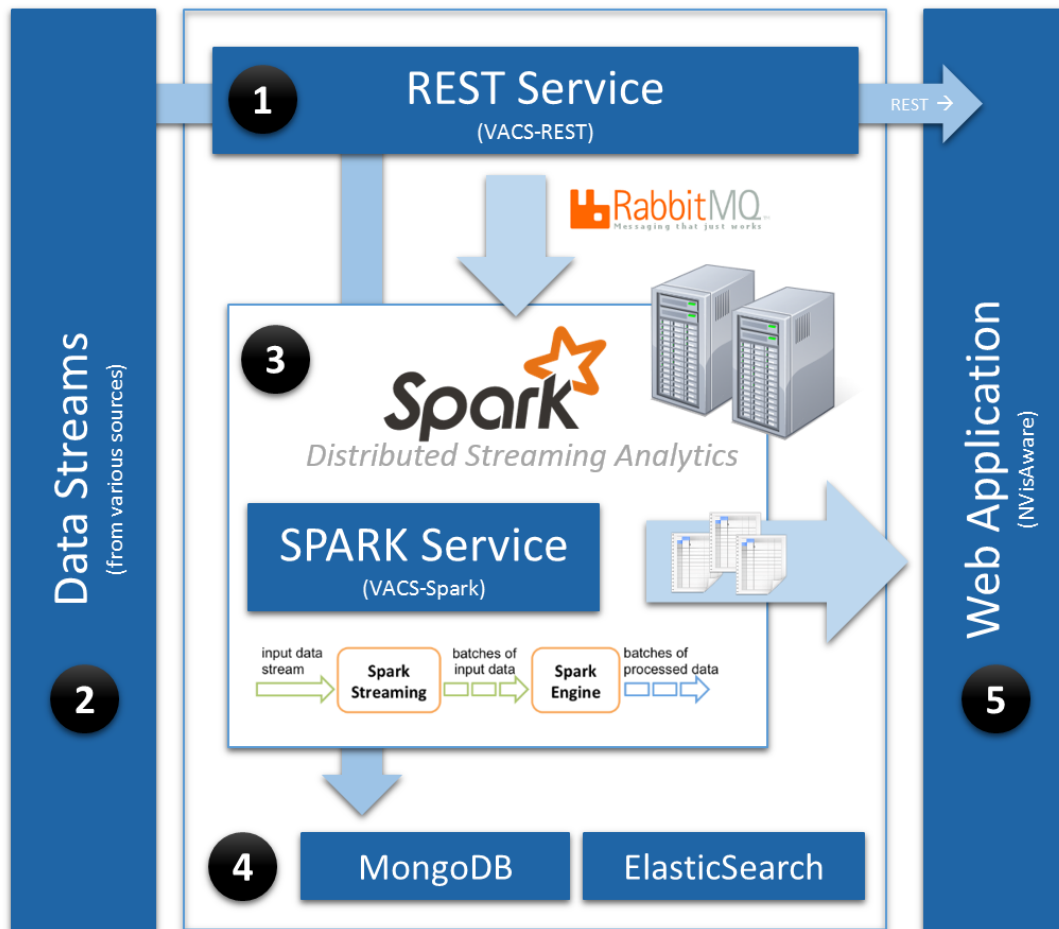
<sup>2</sup> <https://spark.apache.org/>

<sup>3</sup> <http://www.rabbitmq.com/>

<sup>4</sup> <http://www.elasticsearch.org/>

<sup>5</sup> <http://www.mongodb.com/>





▲ **Figure 5.2** — System architecture of *NStreamAware*. Various modern systems, including *Apache Spark*, *RabbitMQ*, *MongoDB*, and *ElasticSearch*, to provide the needed scalability for an interactive visual analytics application for big data use cases.

### REST Service Module

The *REST Service* (1), which is implemented as multi-threaded standalone Java application, provides a REST interface accessible by all other modules, especially the web application. This REST service is used to handle job queuing and to answer data requests. To attach new data streams, the respective jobs can be sent to the service via a defined REST API. The job is added as new thread and the API can be used to control or retrieve status information about these running jobs. Incoming messages from the data stream are then preprocessed, fields are extracted, and eventually treated as individual events, enriched with various additional attributes. The procedure is based on the assigned scenario configuration. For social media messages, sentiment values are calculated, while for IP-related data geo lookups can be made. In practice, many servers do not provide very accurate timestamps, therefore, a new field with the current timestamp is added as well, to have more accurate timings in cases where a sending host does not make use of the network time protocol or uses deviating time settings.

## Scalable Analytics Module

*Apache Spark* provides distributed memory abstraction, that is fault-tolerant and efficient. This helps to program distributed data processing applications without worrying about fault-tolerance. *Apache Spark* introduces a programming model, called Resilient Distributed Datasets (RDDs), which provide an interface to coarse-grained transformations (e.g., map, group-by, filter, join). The RDDs can be addressed within Scala similar to normal collections, however, they are indeed spread over the underlying cluster machines. If a *transformation* is called on a RDD, the execution is actually done on various worker machines. When an *action* is called (e.g., count), the result is retrieved from all workers to return final results. We use the streaming extension of *Apache Spark* and use the same programming model to analyze data streams in real-time. We define a sliding window and connect to a *RabbitMQ* queue to receive messages forwarded by the *REST Service*. Currently, we defined various feature types to be calculated on the incoming messages: *count*, *set*, *new-set*, *key-value list*, and *key-array list*. All features as seen in Table 5.1 for example belong to one of these message types. After calculating the various features, they are directly stored to a *MongoDB* collection. When all features are ready, the web application is notified via *RabbitMQ* to retrieve the sliding slice content via the REST API using the appropriate database queries. *Count* provides a simple counter of number of messages. A *set* stores a list of unique values occurred within a sliding window, while a *new-set* feature will only include values, which have never been seen in the whole stream before. A *key-value list* can be used to count the number of occurrences for all words to gather a list of frequent words. The *key-array list* can be used to store for each key an array of values. This can be used, for example, to track for each IP address, all used port numbers in the sliding window.

### 5.2.2 Conclusions and Limitations

With the infrastructure of *NStreamAware* it is possible to do data fusion for various heterogeneous data streams in a scalable way using *Apache Spark*. The system is able to automatically generate summaries of a variety of features in a window-based fashion as seen in Table 5.1. For example, having lots of IPS alerts (e.g., as represented in the feature *ossecAlerts*), gives a good hint, that there is something bad going on. Judging them without any context is challenging. However, the sliding slice also contains other features from other data streams (e.g., NetFlow details) correlated over time, that might reveal more insights about the actual incident occurred in the current time window. The main limitation of this approach is, to define suitable window sizes. In practice, we tested various window sizes, however it is hard to define them automatically. Good window sizes strongly depend on the data stream characteristics, but also on the user's individual analysis goals, with respect to use cases and detailedness. We leave this particular challenge of defining – or even integrating adaptive sliding windows – for future research. While a tabular representation of the heterogeneous features helps a lot, it is not suitable for quick understanding and timely situational awareness. Therefore, the next section proposes a technique called *NVisAware* to visualize sliding slice data.

▼ **Table 5.1** — **Features for various network-related data streams.** The table shows a selection of aggregation features, which are automatically calculated in regular intervals by our implemented analysis and aggregation module for each sliding window.

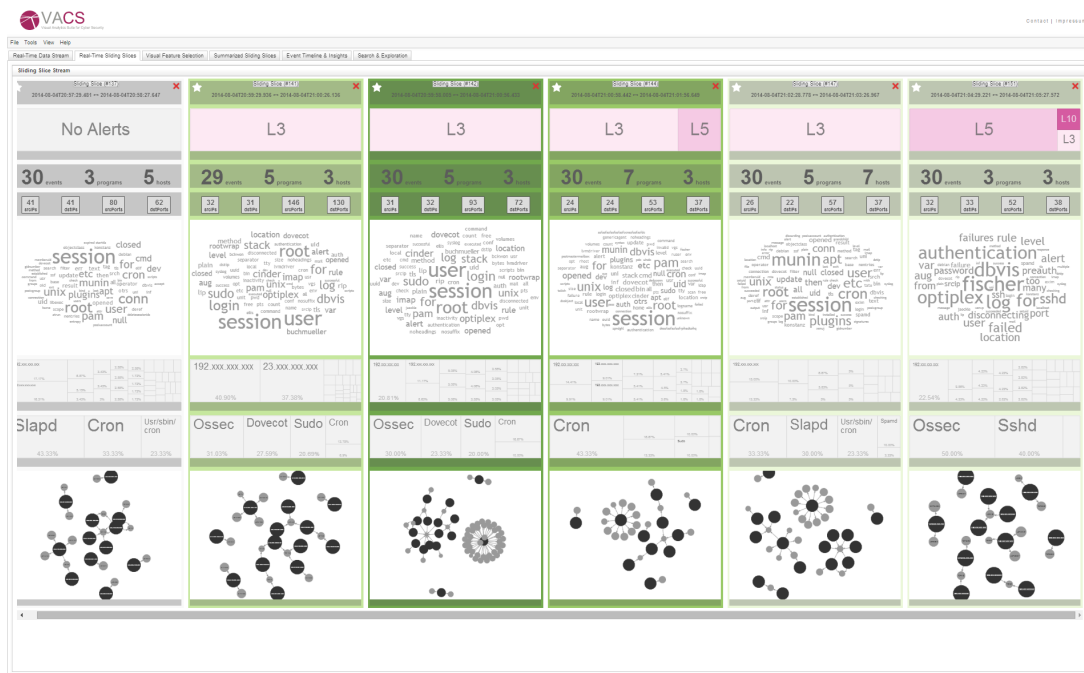
ID	Feature	Type	Stream	Description
(1)	<i>#events</i>	count	Syslog	number of syslog messages
(2)	<i>timestamps</i>	set	Syslog	a set with timestamps
(3)	<i>#programs</i>	count	Syslog	number of programs having syslog messages
(4)	<i>#hosts</i>	count	Syslog	number of servers sending syslog messages
(5)	<i>#frequentWords</i>	count	Syslog	number of frequent words
(6)	<i>programs</i>	key-value list	Syslog	list of programs with respective counts
(7)	<i>hosts</i>	key-value list	Syslog	list of servers with respective counts
(8)	<i>frequentWords</i>	key-value list	Syslog	list of frequent words with respective counts
(9)	<i>newHosts</i>	new-set	Syslog	hosts not seen before
(10)	<i>newPrograms</i>	new-set	Syslog	programs not seen before
(11)	<i>srcAddr</i>	key-value list	NetFlow	list of source IPs with respective counts
(12)	<i>dstAddr</i>	key-value list	NetFlow	list of destination IPs with respective counts
(13)	<i>srcPorts</i>	key-value list	NetFlow	list of source ports with respective counts
(14)	<i>dstPorts</i>	key-value list	NetFlow	list of destination ports with respective counts
(15)	<i>topTalker</i>	key-array list	NetFlow	communication patterns of top talkers
(16)	<i>#srcAddr</i>	count	NetFlow	distinct count of source IPs
(17)	<i>#dstAddr</i>	count	NetFlow	distinct count of destination IPs
(18)	<i>#srcPorts</i>	count	NetFlow	distinct count of source ports
(19)	<i>#dstPorts</i>	count	NetFlow	distinct count of destination ports
(20)	<i>ossecAlerts</i>	key-value list	OSSEC	list of IPS alerts and respective counts

## 5.3 Visual Exploration for Sliding Windows

The graphical user interface provided by our web application, contains various displays. The application is written in HTML5 and JavaScript using various visualization libraries. The display consists of multiple configuration and parameter views and six main tabs: *Real-Time Data Stream*, *Real-Time Sliding Slices*, *Visual Feature Selection*, *Summarized Sliding Slices*, *Event Timeline & Insights*, and *Search & Exploration*. The first display has been described in Section 5.1.1 and is used to take a look at the raw messages in the data stream to get a first overview. In the following section, we introduce the main contribution, which is a widget-based visualization technique for sliding slices.

### 5.3.1 NVisAware – Visualization Technique for Sliding Slices

To visually represent the generated sliding slices, we provide a novel visualization with various embedded charts like word clouds, node-link diagrams, geographic maps, treemaps, and counters within each slice. The slices are juxtapositioned next to each other to provide a timeline based on consecutive slices as seen in Figure 5.3. The prominent background color uses a colormap from dark green over white to magenta based on a diverging *ColorBrewer* [31] set. The color indicates a similarity score to the previous slice to alarm the analyst. In the upper left corner a star icon can be



▲ **Figure 5.3** — Using *NVisAware* to visualize heterogeneous data streams. The figure shows various real-time sliding slices. New slices of the most recent sliding window will automatically added on the right.

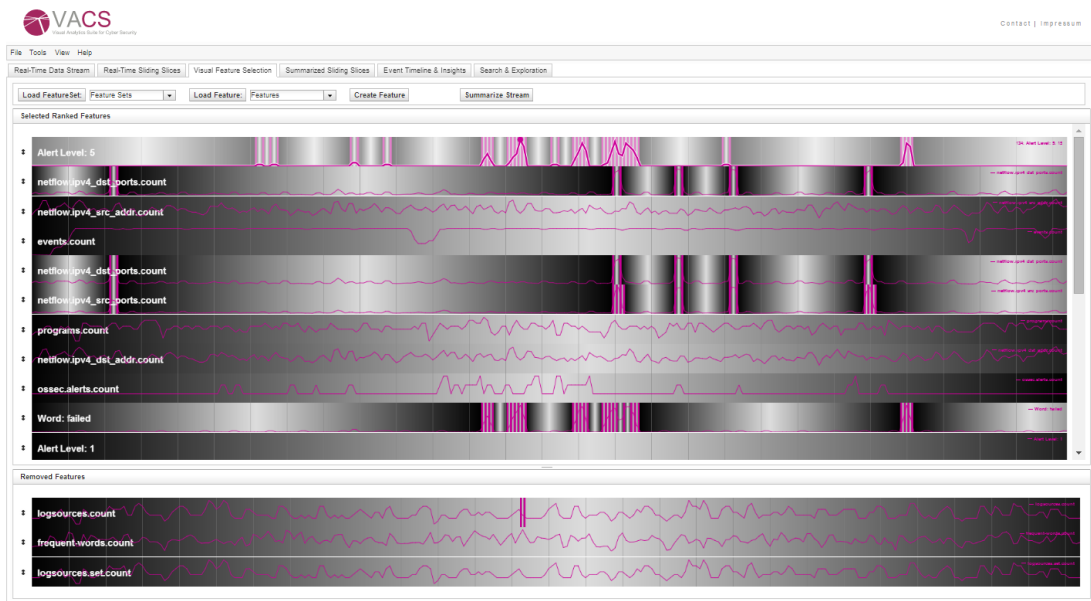
used, to store the slice for further investigations. The slice will also be added to the *Event Timeline & Insights* view, where all starred objects are presented in a traditional interactive timeline to explore the events flagged and labeled by the analysts.

### Real-Time Sliding Slices

As soon as a new slice is available via *NStreamAware*, it is loaded and shown in all open and active instances of *NVisAware* as seen in Figure 5.3. New slices are added on the right, so the analyst can still focus and explore previous slices, while smoothly switching to the most recent one. This helps to reduce the need to stop working on a slice, as soon as new data comes in. The level of stress, and the cognitive load of the analyst can be reduced through the fixed interval, in which new slices become available. In times, when a vast amount of events are suddenly occurring, it won't negatively influence the analysis flow of the user. In such times, a given slice would just contain more events than usual, which would be clearly visible in the respective counters. The top-most widget of each slice, as seen in the example in Figure 5.3, shows the distribution of IPS alerts using their severity identifier (L1-L10) as treemap. Further drill-down interaction would actually reveal the underlying IPS alerts with the respective severity level. Various counters are shown to reflect statistical information about the underlying events. The word cloud represents frequent words based on the occurred syslog messages, while the treemap underneath shows the top talker in the current time window, based on the correlated NetFlow data. The most frequent programs reporting syslog messages are also shown as treemap, while a node-link diagram is used to make the dominating communication patterns (source IPs to destination ports) visible.

### Using Visual Feature Selection for Stream Summarization

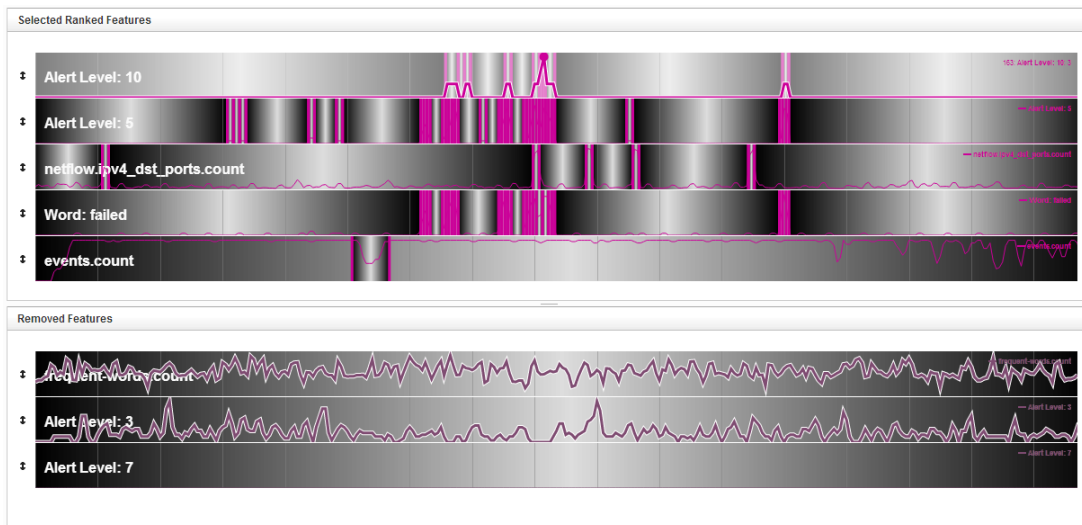
In many situations, the analyst is not interested in following the data stream in real-time. However, in some cases a summary of the current data stream should be provided. Fully-automated summarizations are hard to achieve for complex heterogeneous data streams. Therefore, we provide a visual feature selection interface as seen in Figure 5.4, to steer the merging algorithm based on the user’s criteria.



▲ **Figure 5.4** — Using visual feature selection for *NVisAware*. All temporal features can be visualized as time-series for feature selection.

All *count* features, as for example seen in Table 5.1, can directly be used in the feature timelines in Figure 5.5. More features can be derived from *key-value lists*. For example the occurrences over time of a specific word found in the stream. Each feature timeline contains many values, one value for each sliding slice observed so far. This data is processed on the server side and each feature timeline is cut into segments: Each timeline is clustered using the DBSCAN [72] algorithm. Afterwards, consecutive slices belonging to the same cluster are merged to a segment. The start and end points of these possibly important segments are visible as vertical colored lines and through the background shading within the timelines. The analyst can visually interpret these segments, modify them, or add new segments for interesting parts, which were not detected by the algorithm. The analyst can remove or reorder the features using drag and drop. The final feature order and selection is sent to the REST service, where all segments are merged together with the given constraints, while ignoring low-ranked conflicting features and keeping non-conflicting and more specific segments.

Eventually, the original sliding slices are compressed using map and reduce on the *MongoDB* database according to the heuristic merge and importance model. Less important segments are merged together providing a multi-focal scaling of the data stream steered by the analyst according to the tasks at hand. The list of features is evaluated by the heuristic to ensure: (i) the order of the feature list as defined by the analyst, (ii) if the identified segments in the following features are non-conflicting keep

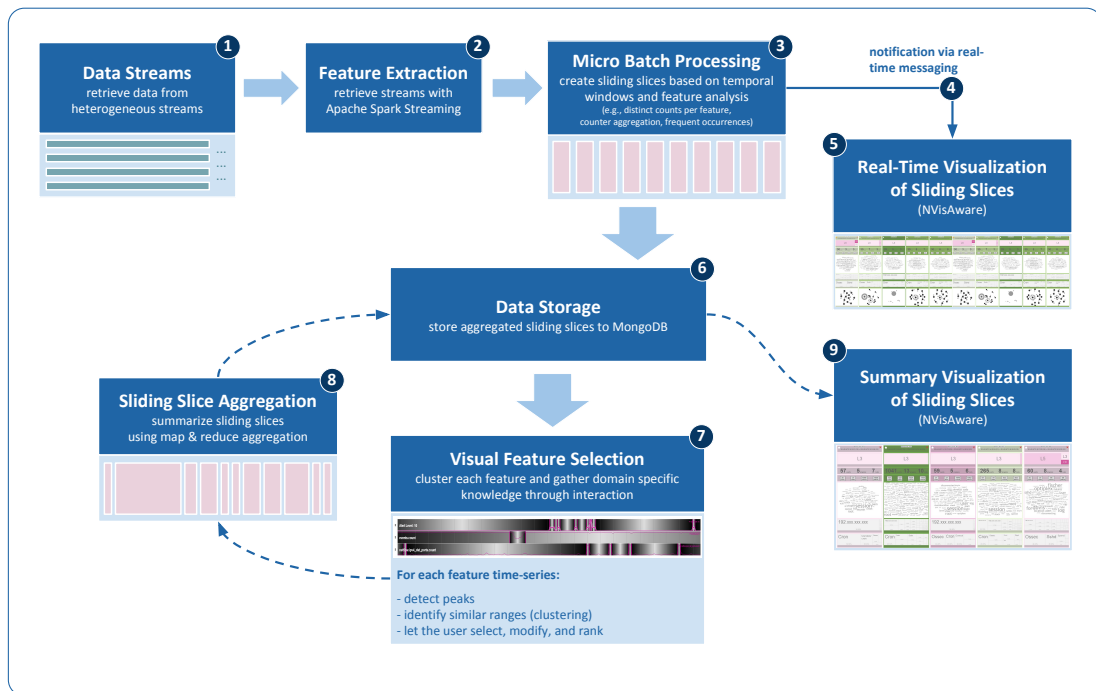


▲ **Figure 5.5** — **Example of selected features.** The analyst is in the loop to steer the merging algorithm through selecting, ranking, and modifying feature segments, to provide meaningful summaries of sliding slices. *Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.*

both, (iii) in case of conflicts, keep more specific segments, (iv) do not overwrite specific segments with more general spans, to (v) include smaller events, when completely within a larger segment.

### Interactive Data Stream Summarization

The interactive analysis workflow is shown in Figure 5.6. Various data streams (1) are eventually retrieved via the *REST Service* and analyzed by the *Spark Service*. Features are extracted (2) and correlated from the heterogeneous streams. With the support of *Apache Spark Streaming*, micro batch processing is used to create sliding slices based on temporal windows to conduct feature analysis. The streams are correlated and aggregated statistics are calculated (e.g., distinct counts per feature, counter aggregation, frequent occurrences) as presented in Table 5.1. If real-time *NVisAware* displays are registered via the message broker, they get notified about the new available sliding slice and the slice is automatically added to the visualization display as seen in Figure 5.3. Additionally, the whole data is sent to the data storage (6) to support interactive data stream summarization, in which the user interacts with the extracted time-series based on the sliding slices (7). Interesting segments are automatically highlighted based on clustering and peak detection. The user can further assist the analysis through domain specific knowledge: The analyst can select relevant features, modify, and rank them. Additionally, the suggested segments can be modified if needed. Afterwards the result is sent back to the server, which summarizes the respective sliding slices using map and reduce aggregation (8), which is executed on the data storage (6). Afterwards, the resulting summarized sliding slices are pushed to *NVisAware*. The visualization then shows summarized sliding slices for the originally defined time span. More slices will be shown for times with lots of interesting segments, while slices belonging to less interesting segments are merged into a common single slice instead.



▲ **Figure 5.6** — Overview of the interactive visual analytics workflow. Visualization with *NVisAware* is used for real-time analysis, but also for summary visualizations based on the multi-focal aggregation of sliding slices using an interactive visual feature selection process.

### 5.3.2 Evaluation using Network Security Case Study

In general, it is quite challenging to evaluate complex visual analytics applications. Individual design decisions can be formally evaluated in user studies and many decisions are indeed based on perception studies. However, proper evaluation of complex expert applications is more than to evaluate all individual design decisions. Describing convincing use cases or presenting case studies with experts are often the only reasonable ways. However, also these results are often subjective and hard to compare to alternative approaches. Another reason is, that “*insight, the major aim of visual analytics, is ill-defined and hard to measure*” [255]. This is even more true, if we are talking about a mental state of situational awareness as goal of the system.

Having this in mind we decided to go for two directions of evaluations. Firstly, we describe a case study, how our system can be used in an operational computer network of a working group to help the system administrator to stay informed about the most important activities. Secondly, to evaluate the real-time capabilities of our system and the insights management, we actively participated in VAST Challenge 2014 with an early version of our prototype.

To show the capabilities of our system, we implemented our system in a computer network of a working group with about 85 active local devices including workstations, mobile devices, and servers, producing about 1.4 million NetFlow records per day with peaks up to 10,000 records per minute. 13 servers are connected to a central syslog server, producing 30,000 to 80,000 messages per day with individual peaks of up to



▲ **Figure 5.7** — Example of findings using *NVisAware*. Visualization to monitor data streams using sliding slices to reveal interesting findings. The interactive display can be explored by the analyst while new sliding slices are continuously added. *Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.*

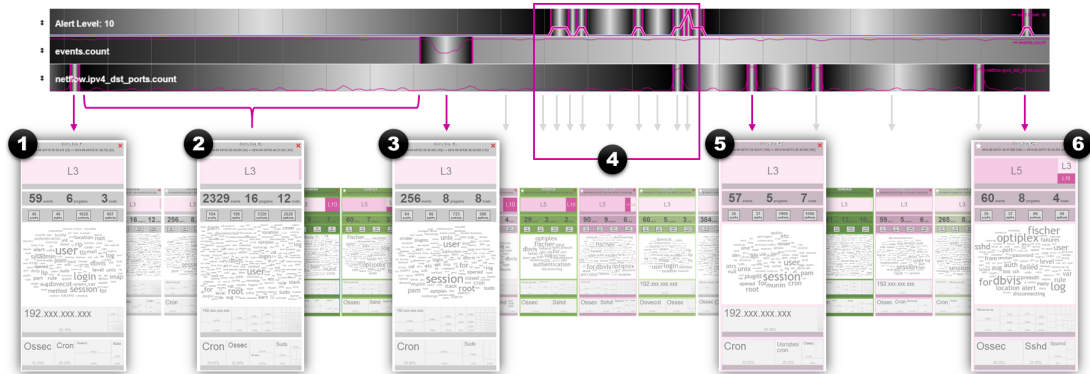
5,000 messages per minute. These servers are also monitored using *OSSEC* [184], which is a widely used “*host-based intrusion detection system that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response*” [184]. The generated alerts are also pushed to the central syslog server. With this infrastructure in place, we were able to forward the data streams to our *REST Service* to make them available for *NStreamAware*. In the following, we made use of the system log stream (SL), NetFlow stream (NF), and OSSEC alert stream (OS). It would be easy, to further include additional data from the underlying network, for example, system metrics, Snort alerts, or web server access logs.

The analyst opened the web application in a modern web browser and added the data streams as jobs to the server-side *REST Service*. Seconds later, the first messages appeared in the *Real-Time Data Streams* tab as seen in Figure 5.1. This view is a split-screen showing the real-time events of SL and OS as textual messages. The bottom window presents a zoomable geographical map to plot and cluster extracted geographic locations. NF records are not plotted to the geographic map, because a geographic map of the total IP traffic will most likely not provide actionable insights. However, mapping specific IP addresses of successful logins can be worth monitoring to identify suspicious behavior or to reveal misuse of login credentials. Furthermore, real-time filtering and search can be applied to reduce the number of live events shown in the display.

The *Spark Service* was operated in local mode on a normal workstation *Dell OptiPlex 980, Core i7-860, 8GB RAM 4x 2.80GHz* with 10 separate working threads. To provide further scalability the service could also be deployed to a cluster of hardware machines running *Apache Spark* or to a cloud-based deployment. To provide a new sliding slice



every 30 seconds, we initialized the system with a batch and slide interval of 30s and a window length of 60s. These settings depend on the general characteristics of the data streams.



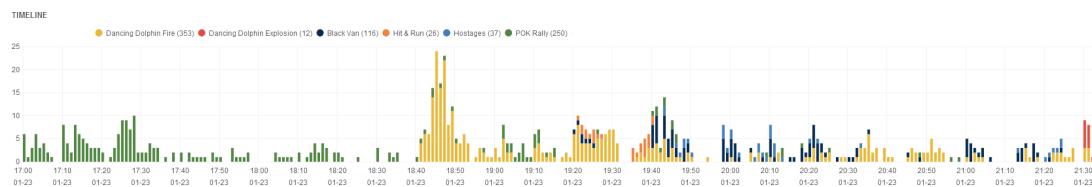
▲ **Figure 5.8** — Example of summarization results. Visual feature selection helps to merge many slices to single summary slices. *Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.*

To reduce the cognitive load, the analyst decided to switch to the real-time sliding slices visualization as seen in Figure 5.7 showing an example of five consecutive slices. The interactive display can be explored by the analyst while new slices are continuously added to the right in regular intervals to support situational awareness. The first slice in Figure 5.7 contains critical *OSSEC* alerts (L5, L10, L3) visualized in a small treemap widget (1). Alerts with a severity of 10 should warn the analyst of ongoing security issues, which should be explored using drill-down functions. Those alerts are related to authentication issues as seen in the word cloud (2). Another treemap widget in the first slice (3) gives an overview of involved programs. The third slice suddenly reveals a high port usage (4), which can be recognized at the port counter. The treemap of source hosts (5) reveals the source host. The analyst can use the IP-Port node-link diagram based on NF (6) to visually explore those suspicious connections.

Later on, the analyst decided to not look on all sliding slices, but to compress the view based on specific features. Figure 5.8 shows that the analyst is interested in slices with highly critical *OSSEC* alerts of level 10, segments based on the number of syslog messages received, and based on the number of destination ports utilized in the computer network. Based on this selection the slices are merged accordingly. In Figure 5.8, (1) relates to the segments relating to a port scan. After that, there were no important slices according to the feature selection, so a long time span is merged to a single summary slice (2). The analyst was also interested in the message drop in (3). Then various *OSSEC* alerts occurred in multiple sliding slices (4). This area seems to be highly suspicious, leading to many individual summary slices to provide more details. Eventually, there are further suspicious events based on NF data in (5) and another peak with *OSSEC* alerts in (6) related to invalid SSH logins.

### 5.3.3 Evaluation using VAST Challenge 2014<sup>6</sup>

Evaluation of a real-time visual analytics application with respect to situational awareness is challenging and hard to compare. To reuse a given dataset and stream it back to a system under review often cannot provide a faithful evaluation, because the actual dataset is normally known by the researcher, so the implemented tools can be specifically designed for that particular dataset. Known details from previous forensic analyses based on the full dataset also bias proper evaluations. However, in real scenarios, the data – completely unknown to the analyst – comes in *incrementally*, so that patterns and insights evolve over time. Luckily, the VAST Challenge 2014 exactly addressed that problem in Mini-Challenge 3 (MC3) and provided a real-time data stream and forced the participants to actually work on a previously unknown real-time data stream. We, therefore, took this opportunity and actively participated in this challenge to evaluate the applicability of our method to solve the given streaming challenge. The data was not related to cyber security, but contained various social media streams. However, this still enabled us to provide a proof-of-concept for situational awareness in data streams. Furthermore, the successful participation shows the general applicability of our approach, which was as shown in the case study originally developed for cyber security.



▲ **Figure 5.9** — **Timeline of major VAST Challenge 2014 MC3 events.** The colored histogram highlights major events based on extracted keywords and insights of interesting events, which could be identified in real-time. *Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.*

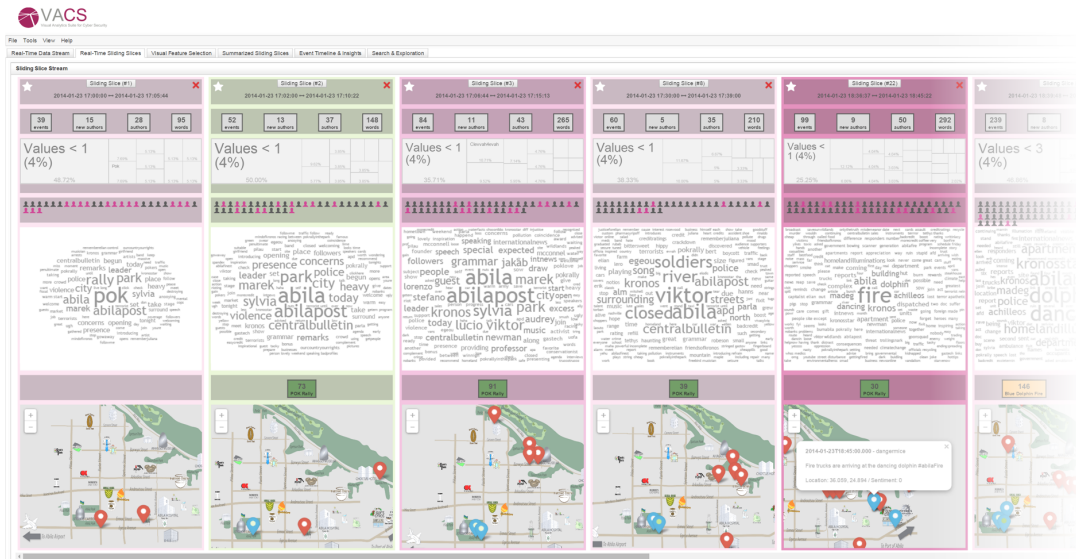
#### Application for Social Media Streams

The requirement of the challenge was to analyze the data streams made available by the organizers over a *WebSocket* connection in real-time. The data stream contained a stream with micro blog and call center messages covering a time from 17:00 to 21:30. A first analysis had to be sent to the committee within three hours after first connecting to the final data stream from 20:00 to 21:30, which could only be streamed once, to force the participants to do real-time processing and provide immediate situational assessment under time pressure.

The fictional but realistic scenario was the so-called *Kronos Incident* in which several employees of a company named *GAStech*, located at the island of *Kronos* went missing. Because of an ongoing conflict between an organization known as the *Protectors of Kronos (POK)*, they are suspected in the disappearance. Within that challenge, the main focus of MC3 was to analyze a real-time data stream based on micro blog records that have been identified by automated filters as being potentially relevant to the ongoing incident and text transcripts of emergency dispatches by the local police and

<sup>6</sup> <http://vacommunity.org/VAST+Challenge+2014>

fire departments. During the real-time analysis, we could identify multiple interesting events using *NVisAware* and could make sense of the overall story, as summarized in Figure 5.9. The first event as showed up in Figure 5.10, which is present from 17:00 until around 19:00, is a rally, organized by POK with different speakers and a concert. Various leaders of POK and many supporting persons are attending this gathering.



▲ **Figure 5.10** — Using *NVisAware* to solve the VAST Challenge 2014 MC3. New sliding slices are automatically added on the right to summarize the most recent events happening on Kronos island.

At one point an incoming sliding slice caught our attention by the red colored background indicating much changes to the previous slice as seen in Figure 5.11. And indeed various people are actively talking about a major fire. Using the geographic map, the location can be easily identified, which was the so-called *Dancing Dolphin Apartment*. After firefighters were able to get the fire under control, we realized subtle messages around 21:00 that the fire flared up again, resulting in an explosion later on around 21:30.

The most important event with respect to the overall situation was an incident related to a black van involved in an hit and run. A biker got hit by the car. In the following, the van was spotted again at another location and could be stopped at *Gelato Galore* by the police leading to a standoff. It turns out that there are two hostages and two kidnappers in the van. In the process one of the suspects shot a police officer. After a while the SWAT team arrived and suddenly the suspects surrendered and could be taken into custody. It turned out that the two hostages were indeed employees of *GAStech*. With this information and insights from the data streams, we were also able to participate in the grand challenge, in which two other mini challenges with additional datasets had to be solved. After combining all the information, we gained a high level of situational awareness and were able to make informed decisions and isolate locations, where the other hostages are likely to be held.



▲ Figure 5.11 — An example for a major fire incident. Three stream slices out of many slices of the data stream to show the evolving topics in the word cloud. The red background color reflects a high word cloud dissimilarity to the previous slices.

### Evaluation Results

VAST Challenge 2014 received a total of 73 submissions, while 23 teams submitted to MC1, 30 to MC2, 13 to MC3, and 7 to the grand challenge (combining all three mini-challenges) [270]. In the history of VAST Challenges, MC3 “was the first to require access and use of streaming data. While first-of-a-kind mini-challenges are often less popular than more traditional mini-challenges” [270] the committee still received 13 submissions. “The VAST Challenge committee recruited reviewers with expertise either in visual analytics or related disciplines and domain experts. Ninety-five reviewers participated, each providing from 1 to 9 reviews. Each submission received 3 to 6 anonymous peer reviews. (...) They were asked to provide an overall rating, comments on the overall rating, a review of how well task questions were answered and how well visual analytics were applied, including whether or not innovative tools were created for the challenge” [270]. Afterwards, the committee “held three separate one-day meetings to determine awards for each of the mini-challenges and grand challenge” [270]. MC3 was especially challenging, because it “required sophistication in both receiving and processing the social media stream as well as analyzing the stream in real-time” [270].

Our submission was reviewed by 4 independent, anonymous, and external reviewers (R1-4). Eventually, our approach got awarded for an “*Outstanding Comprehensive Mini-Challenge 3 Submission*” [270]. In the following, we want to highlight some comments as provided by the anonymous reviewers.

The first reviewer stated: “*This is the best system I have reviewed. The strength is the clarity of presentation, and most important the scalability of the architecture.*” (R1). While R1 focused primarily on the overall system, R2 highlights the visualization in more detail and states, that “*the mixed use of self developed and already existing tools represents a clear strength of the submission, as it shows how a realistic analytic discourse can be created making use of different combined coordinated visualizations*” (R2). Additionally, R2 concludes that “*the proposed approach does not show only the conclusions just to find an answer to the questions of the challenge, but rather illustrates how a complex reasoning could be carried on in a realistic manner and in real time*” (R2). R3 further focused on the usage of our “*useful and interesting method*” (R3) and highlights the fact that it “*significantly reduces the burden on the analyst when first analysing the data*” (R3). However, R3 also concludes that “*further aggregation of text (...) could be incorporated in events message stream to reduce text on the screen*” (R3), which is a valid point. We agree that there are many more possibilities to further enhance the system with respect to text analytics. The reviewers also had to judge identified findings and provided also quite positive feedback: “*The team have successfully found 4 major events*” (R3). Furthermore, the “*slices provide a rich snapshot of the immediate past that has enabled the team to show how they found events in a quick manner*” (R3). R4 states that there “*were some inaccuracies in interpreting the data, but some visualization techniques were quite useful*” (R4). The *NVisAware* visualizations “*were a great approach to showing chunks of real-time data*” (R4). In summary, R4 concluded, that the “*real-time stream analytics (slices) was a novel and visually attractive approach to summarizing chunks of data by time so that they are more consumable*” (R4).

## 5.4 Limitations and Conclusions

Various performance measurements and evaluations [280] showed that *Apache Spark Streaming* is scalable and fault-tolerant. The different database technologies can also scale out horizontally to handle large data volumes. However, the system still needs to be applied to a larger computer network, which is part of the future work. The main limitation with respect to performance and scalability issues could be found in the web application, developed in HTML5 and JavaScript. While the backend is able to use capped collection, and provides data rotation and retention strategies, the real-time graphical user interface does not. When displaying hundreds of sliding slices at the same time the performance decreased, because of browser and memory restrictions of the workstation. However, the responsiveness could be improved by including paging mechanisms in all views and by integrating automatic heuristics to remove old elements. More work needs to be done to keep the display interactive when analysts use the web application for hours without manually removing or reloading some displays.

Automatically defining good sizes for the sliding windows is also planned for the future. The merging model based on the feature selection process, could also be applied to the real-time stream, to actually merge sliding slices in real-time, which is not fully implemented yet. Tracking individual events over time was not the focus of this work, however, more work seems to be promising to extend the approach in that respect

as well. To comply with privacy standards and to prevent misuse of such a system, more research should be conducted to integrate privacy-aware technologies or establish multiple access levels.

Overall, based on various design considerations addressing current limitations of related work, we presented in this chapter a visual analytics system, called *NStreamAware*, to analyze data streams. The interactive user interface is integral part of the whole system, because it allows analysts to interact with the system and steer the clustering process to condense data streams to meaningful segments. The system incorporates novel scalable analytics methods (DC1) through *Apache Spark Streaming*. This enables real-time monitoring with interactive filtering (DC2), which is implemented using *RabbitMQ* as message broker to provide real-time communication between the different modules. Furthermore, we developed a visualization technique to visually represent the generated *sliding slices*, to present data stream summaries using deterministic screen updates using aggregations on sliding windows (DC3). This approach also makes it possible to combine various heterogeneous data sources (DC4) and include them with various embedded visualizations like word clouds, node-link diagrams, treemaps, and counters. To summarize data streams and to provide a multi-scale timeline to compress the stream based on user-steered interesting functions, we integrated an interactive visual feature selection display (DC5). Eventually, we evaluated our system using data streams of an operational computer network and used our system to successfully participate in the VAST Challenge 2014 to compete with other real-time SA solutions and show the applicability for other domains.

*The man who trades freedom for security does not deserve nor will he ever receive either.*

— Benjamin Franklin



## Conclusions and Future Research Directions

### Contents

---

<b>6.1 Summary</b> . . . . .	<b>165</b>
<b>6.2 Contributions</b> . . . . .	<b>167</b>
<b>6.3 Future Perspectives</b> . . . . .	<b>169</b>

---

THIS thesis proposed a variety of different visual analytics methods to enhance situational awareness for cyber security. To achieve this research goal, we defined various research objectives and actually covered more than that, as we eventually contributed to all network security use cases as originally defined by Shiravi et al. [216]. This chapter summarizes the work of this dissertation, reviews the main contributions, and highlights various important future research directions.

### 6.1 Summary

After introducing the overall motivation for the urgency of improving situational awareness for cyber security to sustain future attacks in Chapter 1, it became clear that visual analytics is a suitable approach to address this challenge. In Chapter 2, we conducted an extensive literature review focusing on visualization systems supporting situational assessment in cyber security. The literature review incorporates existing related surveys, but also includes novel methods of recent years, which have not been covered by existing surveys, but could be identified using the systematic review methodology. Afterwards, we discussed our observations and revealed various research gaps. To convey the broad field of cyber security, we decided to structure the thesis according to major visualization use cases for cyber security.

Chapter 3 focused on monitoring of network activity. We primarily introduced VACS, which is a web-based visual analytics suite for cyber security, originally designed

for internal and external monitoring of computer networks (Section 3.1). Because of the open challenge to properly evaluate complex visual analytics applications, we actively participated in international competitions to evaluate and compete with others on realistic datasets, for which ground truth data is available and judgment is made by independent reviewers, who are either visualization or domain experts. Their feedback helped us to address limitations of *VACS* with respect to scalability and context-awareness. We, therefore, integrated more analytical methods to enhance visual correlation for port activity monitoring in Section 3.2 and proposed a novel visualization technique, called *ClockMap* in Section 3.3. This scalable approach, which is a novel combination of circular temporal glyphs and radial treemaps, filled an open research gap to analyze temporal network activity with respect to a given hierarchical context. We, and also other researchers, extensively evaluated this approach from various perspectives and we were able to successfully identify almost all events of the VAST Challenge 2013 ground truth data.

While it was possible with *ClockMap* to identify suspicious hosts and servers within computer networks, we shifted our focus specifically to the visual analysis of various network threats in Chapter 4. While most of the research in cyber security visualization in the last decades has focused on techniques to visually represent attack patterns, we gave a brief overview about the state-of-the-art in Section 4.1 and focused primarily on evaluating a novel generic approach, called temporal multi-dimensional scaling (TMDS) using a network security case study and the VAST Challenge 2013 dataset. Less research has been conducted focusing on a major cyber security threat, called IP prefix hijacking, which we addressed in Section 4.2. Such attacks targeting the border gateway protocol (BGP), which is crucial for routing in the Internet, have severe consequences. Together with security experts from Symantec we, therefore, proposed a novel visual analytics system, called *VisTracer*, to correlate routing anomalies based on traceroutes with ongoing spamming activity by attackers. Eventually, we evaluated our approach with cases studies, provided by Symantec's security experts who used *VisTracer* to successfully identify various IP prefix hijackings. Another major threat in the cyber world, are advanced persistent threats (APTs), which often involve highly specialized malware samples. Because of the high relevance, and the emerging visualization techniques proposed in the last years, we introduced a taxonomy of visualization systems for malware analysis in Section 4.3. Furthermore, we not only summarized the state-of-the-art, but also identified future research directions of visual analytics for malware behavior analysis. However, security analysts are not only interested in the detailed analysis of individual malware samples, but also on the threat landscape on a larger scale. Attributing attacks and involved malware samples to related attack campaigns is crucial for situational awareness with respect to the modus operandi of groups of attackers. We addressed this challenge related to strategic decisions in Section 4.4, in which we implemented various visualization techniques into *VACS* to analyze the result of recent clustering algorithms specifically designed for threat intelligence. To evaluate our approach, we conducted a field experiment in the premises of Symantec security response with leading cyber security experts.

The literature review showed, that most of the visual analytics techniques in cyber security, do not explicitly focus on the dynamic real-time characteristics. However, with respect to situational awareness, real-time capabilities are crucial. To emphasize the importance and foster more research in this direction, we introduced a novel and scalable infrastructure, integrated to *VACS*, for heterogeneous network stream analysis in Chapter 5. We specifically proposed, *NStreamAware*, which is a stream analysis



system based on *Apache Spark* to provide aggregated data slices to be presented to the analyst using a novel visualization technique, called *NVisAware*. Furthermore, we integrated visual feature selection techniques to provide meaningful summaries of those slices. Eventually, we successfully evaluated the system using a network security case study, and evaluated the general applicability in the context of situational awareness through active participation in VAST Challenge 2014, in which our approach got awarded for an outstanding comprehensive submission.

Overall, we successfully fulfilled the stated research goal and proposed, implemented, and evaluated interactive visualization systems to enhance situational awareness in cyber security through the scalable exploration of network activity, the analysis of network threats, and visual analytics support for the analysis of heterogeneous data streams by combining automated methods with scalable and interactive visualizations.

## 6.2 Contributions

In the following, we briefly summarize the main contributions, as provided in this thesis, which formed the basis for successfully addressing the overall research goal of this work.

### I. A Survey of Visualization Systems for Cyber Security

In this thesis we contributed a comprehensive survey of visualization systems for cyber security. This was the first literature review covering the broad field of cyber security visualizations, including an extensive survey of BGP, malware, and attack attribution visualization systems with network security applications, extending existing taxonomies to provide a holistic view of visualization methods to enhance situational awareness.

### II. A Taxonomy of Visualization Systems for Malware Analysis

Additionally, we provided together with Wagner et al. [261], the first survey and taxonomy of visualization systems for malware analysis, which is an emerging field of research in the last years. Together with malware security researchers, we extensively analyzed the various capabilities and classified them according to major use cases. Furthermore, we proposed various future research directions to guide other visual analytics researchers to interesting starting points and research gaps to further enhance malware analysis through novel visualization approaches.

### III. ClockMap – A Visualization Technique for Scalable Exploration of Hierarchical Temporal Data

The main contribution of *ClockMap* was the novel combination of clock-based glyphs with circular treemaps. Although, there are major drawbacks of such treemaps, we showed in various case studies, that the integration as layout algorithm for the placement of circular glyphs is quite effective for situational awareness in cyber security especially with respect to host and server monitoring. This technique was further evaluated and integrated into *BANKSAFE* [90] to participate in an international visual analytics competition to solve realistic cyber security tasks. Furthermore, we evaluated various design decisions together with Fuchs et al. [95] using formal user studies. This contribution primarily fulfills RO1, as described in Section 2.3 to “introduce novel visual techniques for context-aware exploration to support visual analytics for network activity” [page 37].

#### IV. IAS-Explorer – A Visual Analytics Method for Privacy-Aware Correlation of Temporal Network Activity

The main contribution of *IAS-Explorer* in the scope of this thesis was a visual analytics method to analyze, visually explore, and monitor large numbers of network activity time-series. With this, we were able to support port activity use cases especially for privacy-preserving setups, and to provide better means to correlate network activity time-series. Our approach using vertically arranged line charts in combination with integrated analytics to retrieve similar time-series effectively avoids overplotting and forces the user to focus on the comparison and correlation of specific patterns and temporal network activity peaks. This contribution partially fulfills RO1, as described in Section 2.3 to “*introduce novel visual techniques for context-aware exploration to support visual analytics for network activity*” [page 37].

#### V. VisTracer – A Visual Analytics System for BGP Prefix Hijacking Detection using Traceroutes and Spam

The novel visual analytics system, *VisTracer*, combines IP traceroutes from ongoing spam and phishing campaigns to correlate BGP routes with malicious network threats. In particular, we contributed a visual analytics tool to analyze traceroute data, provided a successful integration into a large-scale automatic analysis system, and introduced novel glyph- and graph-based summary visualizations for routing data. Furthermore, we evaluated the approach together with experts and shared the results of their case studies of suspicious routing anomalies with respect to spam activities to identify BGP prefix hijacking. This contribution primarily, fulfills RO2, as described in Section 2.3 to “*combine multiple data sources to improve SA for BGP routing*” [page 37].

#### VI. NStreamAware – A Visual Analytics System to Enhance Interactive Analysis of Heterogeneous Data Streams

The main contribution of this work was a system architecture, called *NStreamAware*, based on *Apache Spark* to summarize incoming data streams in *sliding slices*. Secondly, a web-based user interface, called *NVisAware*, using a novel combination of various visualization techniques to present actionable visual representations of sliding windows to the analyst in real-time to convey the current state based on correlated heterogeneous data streams. Furthermore, we integrated interactive visual feature selection techniques, to provide meaningful summaries of those slices. This eventually produced context-aware summarizations, steered by the expert knowledge of the analyst to provide scalable methods for long time spans. This contribution primarily, fulfills RO4, as described in Section 2.3 to “*introduce a novel dynamic visualization concept for scalable real-time monitoring for heterogeneous data streams*” [page 38].

#### VII. Evaluation of Alternative Visualizations for Attack Attribution

In the field of attack attribution, we integrated various alternative visualization into *VACS* and conducted a field experiment with cyber security experts in the premises of Symantec Security Response. The analysts made use of our visualizations for visual exploration of clustering results generated by one of the leading algorithms for attack attribution [249]. The usage of this visual analytics system on their own data was very well received and they provided extensive feedback which will be valuable for future

research in visualization support for threat intelligence. This contribution primarily, fulfills RO3, as described in Section 2.3 to “*integrate visual analytics techniques for attack attribution*” [page 37].

## VIII. VACS – A Visual Analytics Suite for Cyber Security

The main contribution of *VACS* was originally the incorporation of various common visualization techniques for internal and external monitoring of computer networks. However, in the context of this thesis, we integrated many more of our proposed methods and implemented them on top of *VACS*. This eventually lead to a research prototype, providing a visual analytics suite for cyber security with respect to network activity, threat, and data stream analysis use cases.

## IX. Evaluation of Security Applications through Competitions

Eventually, we extensively showed in the aforementioned systems how active participation in international competitions can help to successfully evaluate complex security-related visual analytics applications. We reported not only the results but also shared feedback and limitations addressed by the anonymous reviewers. We also showed how other challenges (e.g., VAST Challenge 2014) not focusing on cyber security, could still be used for the evaluation of real-time aspects which are comparable to 24/7 monitoring in cyber security. We also made use of our knowledge about these challenges to help other researchers evaluating their approaches (e.g., Jäckle et al. [133], Behrisch et al. [20]).

## 6.3 Future Perspectives

Besides of detailed future research directions with respect to particular cyber security use cases, as discussed in various sections of this thesis, we briefly want to highlight some more general future perspectives to conclude this thesis.

### Implications for Privacy

In the context of this thesis, we deliberately haven’t discussed the implications on privacy of this work. We only partially addressed this sensitive issue in Section 3.2.1 in the context of describing *IAS-Explorer*, which is a privacy-preserving monitoring system. However, most of the other techniques especially in Chapter 3 are not. However, we strongly believe that more research is needed addressing the question how to balance cyber security situational awareness and privacy in an appropriate way. On the one hand, system administrators, network operators, and other security response analysts definitely need novel technologies to monitor and protect the computer networks, for which they are responsible. However, most of such visual analytics technologies can also be used, either on a larger scale to monitor individual (end-)users, or to be used in an unauthorized or unlawful way. A proper discussion of these moral and ethical implications is out of the scope of this thesis. However, like most novel technologies, also visual analytics systems for cyber security can be used for the good, or for the bad. Illegitimate mass surveillance of citizens without proper legislative authorization is an example, for which such technologies are being misused. More research should be conducted, to implement standards, suitable laws, and technologies to protect the privacy of end-users, but still being able to protect computer networks and critical

infrastructures, without the need for detailed large-scale preservation and retention of personal end-user data to eventually respect human rights to privacy and informational self-determination.

### Data Provenance and Uncertainty

The goal for visual analytics applications is to acquire new knowledge about a situation under investigation. The combination of analysis with a variety of visualization tools helps to identify findings and insights, which eventually guide the decision-making process. However, these conclusions can only be as good as the initial data provided by the various data sources. Such issues inject various uncertainties in the workflow. However, it is still an open challenge how to automatically identify such possible uncertainties and how to convey data quality, data provenance, and uncertainty in a way, that the security analyst is not biased or misled to rash decisions. During active participation in the competitions, we were often required to rate the certainty and relevance of stated hypotheses. Thus, handling uncertainty is not only relevant in the context of the input data, but also in reporting findings. The analyst should not only be guided to annotate identified findings, the system should also provide means to relate and link various evidence (e.g., alternative visualizations, or underlying data records) to objectively support stated hypotheses. This research also leads to the next future perspective about communicating and reporting of insights and findings.

### Communication and Reporting of Insights and Findings

Highly interactive visualizations, which are primarily used to identify suspicious events, are often not the best way to communicate and report the acquired insights and findings. However, visual analytics systems should not only provide means for monitoring or threat analysis, but also incorporate appropriate ways to communicate the achieved mental state of situational awareness to others. Reporting could benefit from novel visualizations specifically designed to convey complex findings together with appropriate evidence to allow others to follow the reasoning process of the actual analysts and come to own conclusions and decisions. Only very few existing systems focus on such reporting mechanisms to interactively support the creation of executive or detailed summaries for the various stakeholders.

### Further Integration of Novel Analytics Approaches

The overall goal of the visual analytics approach is a strong and tight coupling of advanced analysis techniques with novel visualizations to eventually combine the best of two worlds: The highly scalable computing power of computer systems and the creativity, intuition, and domain knowledge of the human analyst. However, fulfilling this vision comprehensively is still extraordinary challenging. Within this thesis, we could only provide some steps into the right direction, however more research is needed to tightly integrate advanced analytics, in which the user is able to intervene, re-iterate at various steps and different levels of the analysis pipeline. Developing such holistic visual analytics systems in the field of cyber security requires the close collaboration of experts and researchers with highly specialized in-depth knowledge. This observation makes the following aspect even more crucial for the future.

### Encourage Interdisciplinary Research Projects for Cyber Security

Without a doubt more interdisciplinary cyber security research will be needed in the future to keep on top of the ever growing number of highly advanced threats. More and more devices and systems are added to our computer networks, especially in the context of Internet of Things (IoT). Furthermore, the accessibility of critical infrastructure over the Internet and the deployment of smart grids, will increase the risk and in particular the impact of cyber attacks targeting such systems. To address these challenges and to be prepared for the future, interdisciplinary research is needed to bring together leading experts of various fields. The research in the scope of this thesis was only possible, because of the funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257495, "*Visual Analytic Representation of Large Datasets for Enhancing Network Security*" (VIS-SENSE), which made it possible to combine the strengths of visual analytics and the in-depth knowledge of leading cyber security experts. We, therefore, believe that funded projects, which involve visual analytics researchers and partners from cyber security industry are inevitable for providing novel technologies to continue protecting our connected world.



# List of Figures

## Chapter 1

- 1.1 **Overview of thesis structure.** After the introduction, Chapter 2 presents an extensive literature review in the field of visual analytics for cyber security with a focus on situational awareness. Chapter 3 focuses on visual analysis of network activity, while Chapter 4 focuses on network threats explicitly. Chapter 5 tackles the real-time challenge for situational awareness on heterogeneous data streams. Chapter 6 concludes the thesis and summarizes the contributions. . . . . 4

## Chapter 2

- 2.1 **Methodology of literature review.** The literature review is based on a combination of papers identified within existing surveys and keyword search in various digital libraries to include recent state of the art. . . . . 12
- 2.2 **A survey of visualization systems for cyber security.** An extensive web-based literature review of visualization systems for cyber security. . . . . 13
- 2.3 **Overview for stages of situational awareness.** The general relations between *visualization usage*, *analysis type*, and *stage of situational awareness*. Table 2.3 gives an overview of trends for the various categories. *This modified and adapted figure is based on D’Amico and Kocka [58].* . . . . . 16
- 2.4 **Visualization of attack patterns with *NFlowVis* [81].** The treemap represents the local computer network with hosts as rectangles. External attackers are shown as colored circles on the outside. The splines represent the connections between attackers and computers within the network. This reveals a network scan (from top) and a distributed attack (bottom) originating from hundreds of hosts. . . . . 21
- 2.5 **Propagation flow for BGP update messages.** An example of a BGP announcement originating from AS553 to neighboring AS routers and incremental propagation to ASes around the world. . . . . 25
- 2.6 **Thesis structure based on cyber security use cases.** Visual structure overview of the main chapters along the various network security use cases. . . . . 38

## Chapter 3

- 3.1 **Overview of TCP/IPv4 packet headers.** The figure shows the header fields for an IPv4 packet and the encapsulated TCP segment. . . . . 42
- 3.2 **Dashboard example in *VACS*.** Some widgets on a dashboard showing the current situation using bullet graphs and a temporal histogram. . . . . 44
- 3.3 **Example of interactive line charts in *VACS*.** Interactive line charts show the overall incoming and outgoing network traffic (number of flows). Different normalizations help to focus on peaks or low-traffic periods. Five enormous peaks are standing out in this example. . . . . 44

3.4	<b>Example of striped thumbnail glyphs in VACS.</b> Small multiple visualization of time-series metrics for internal hosts using striped thumbnail glyphs. . . . .	45
3.5	<b>VACS shown on a large powerwall display.</b> VACS can be used on a large powerwall display. Several time-series are shown as interactive line charts on the top to select the overall time window. The colored striped thumbnail glyphs on the left represent the different traffic patterns for relevant internal network hosts. After selection of internal hosts, the interactive node-link diagram displays aggregated connections between different source and destination ports or other external hosts. . . . .	46
3.6	<b>Example of an interactive node-link diagram in VACS.</b> An interactive node-link diagram helps to analyze the aggregated connections between hosts and ports. . . . .	47
3.7	<b>Treemap overview of involved hosts in VACS.</b> A treemap is loaded with underlying data from the selected time span. This helps to identify the top talkers (e.g., IP address with most activity in that time) or to get an overview of involved ports. . . . .	47
3.8	<b>Temporal overview of VAST Challenge’s network flow dataset.</b> Overview of network flows as line chart using square-root normalization for the whole time period. This reveals huge data peaks, which make the analysis of subtle signals challenging. . . . .	49
3.9	<b>Overview of incoming network connections in VACS.</b> As seen in the timeline at the top, almost all network traffic on 2013-04-01 happened in the morning, while there are few interesting peaks throughout the day. The node-link diagram reveals that the majority of the traffic relates to port TCP/80, referring to web browsing by staff members and web server responses to customers. . . . .	50
3.10	<b>Interactive node-link diagram of port scans.</b> The interactive node-link diagram shows port scans with distinct patterns from 10.6.6.6 (Event 3) and 10.7.7.10 (Event 4) to primarily port TCP/80 and TCP/25. The visualization maps the source IP address (10.7.7.10) to a green circle, the lines to many magenta circles refer to the used source port addresses, the lines from those source ports all go through the light-green destination port (TCP/25) to various targeted web servers. The layout is calculated using a force-directed graph layout. . . . .	52
3.11	<b>Treemap representation to analyze DoS traffic.</b> A denial of service (DoS) attack on 2013-04-02 between 05:10 and 07:10 (Event 5). The treemap obviously indicates that most traffic relates to port TCP/80 (HTTP traffic). . . . .	53
3.12	<b>Timline view for a DoS attack.</b> A denial-of-service (DoS) attack on 2013-04-03 between 09:30 and 12:00. The blue line represents traffic on TCP/80, while the yellow line represents traffic on all other ports. This highlights that most traffic involves TCP/80 traffic. . . . .	54
3.13	<b>Using the treemap for root cause identification.</b> Two major port scans from 10.9.81.5 and 10.10.11.15 on 2013-04-06 between 11:10 and 12:00 (Event 12 and 13). The treemap helps to identify the attack origins. . . . .	54
3.14	<b>Visualizing botnet communication.</b> periodic botnet communication over SSH (TCP/22) starting at around 08:20 on 2013-04-12 (Event 23). . . . .	55



3.15	<b>Various sources of port scans on 2013-04-14.</b> Identification of attackers orchestrating various port scans (Event 28) from 10.13.77.49, 10.138.235.111, and 10.6.6.7 starting around 12:20. . . . .	56
3.16	<b>Identification of most attacked ports on 2013-04-15.</b> Various port scans on port TCP/3389 and TCP/25 from 10.13.77.49, 10.138.235.111, and 10.6.6.7 starting around 07:45 (Event 29). . . . .	56
3.17	<b>The main interface of <i>IAS-Explorer</i>.</b> On the left side the data management and time-series selection window is shown. In the main area various visualizations can be shown (e.g., the <i>Explorer View</i> ). On the right side the user can adjust settings to fully configure the visualizations. <i>Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	62
3.18	<b>Explorer view showing 63 different time-series.</b> Each series is scaled in a way, that the data of all time-series fit on a common workstation display. The order of the time-series plots is determined by the volatility of the data in the focus area. <i>Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	64
3.19	<b>Overview of time-series for various descriptors.</b> The <i>Explorer View</i> showing the time-series selected at the beginning (the first six) and the time-series returned by the server by the similarity query (the last seven). <i>Reprinted from [226]. © 2013 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	66
3.20	<b>Interaction workflow of <i>IAS-Explorer</i>.</b> This interactive process when using <i>IAS-Explorer</i> shows the combination of user interaction and analytical guidances by the underlying models. . . . .	68
3.21	<b>Conceptual design of a clock glyph.</b> Visual representation of a single clock glyph, also named <i>clockeye</i> , showing a time-series of 24 hours. Each one hour sector is colored by its data value. Circular shading is applied to emphasize the borders of the glyph. <i>Reprinted from [82]. © 2012 The Eurographics Association.</i> . . . . .	72
3.22	<b>Expanded overview of a whole circular treemap in <i>ClockMap</i>.</b> A circular treemap is used to lay out hundreds of <i>clockeyes</i> into groups based on their hierarchy. The rectangle illustrates, how the visualization will look like, when the user zooms out. <i>Reprinted from [82]. © 2012 The Eurographics Association.</i> . . . . .	73
3.23	<b>Interaction workflow with <i>ClockMap</i>.</b> The interactive workflow in <i>ClockMap</i> uses semantic zoom to enhance monitoring and investigation of hosts in a computer network. . . . .	74
3.24	<b>Hosts and servers within an interesting subnet.</b> Underlying hosts of a very prominent subnet having no night time traffic identified using the interactive workflow represented in Figure 3.23. . . . .	75
3.25	<b>Comparing rectangular versus circular treemap.</b> Side-by-side comparison of a circular treemap with clock glyphs and a rectangular treemap with embedded striped thumbnail glyphs. The temporal locations are better conveyed in the circular representation, because of the stable aspect ratios compared to the rectangular example. . . . .	76

3.26	<b>Evaluation for peak comparison on temporal locations.</b> The summary of results for our conducted user study. It reveals differences between various glyph designs with respect to accuracy, efficiency, and confidence. The clock glyph had relatively high accuracy, with good efficiency (low completion time, especially for low densities), and achieved high confidence scores for temporal peak locations tasks. . . . .	80
3.27	<b>BANKSAFE in a control room scenario.</b> The usage the system in a control room setting helps to analyze big data in large-scale computer networks to achieve situational awareness. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	83
3.28	<b>Point-in-time network health overview.</b> This treemap visualization provides an overview for the current state of the whole computer network. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	84
3.29	<b>Network health overview during an infection.</b> This case represents a wide-spread computer infection leading to a high percentage of critical policy levels. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	84
3.30	<b>Treemap to convey a facility’s policy distribution.</b> A treemap visualization showing the percentage of computers with different policy levels in a single facility and region. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	85
3.31	<b>Activity-policy matrix visualization.</b> Colored rectangles represent the number of hosts having a particular activity flag and policy combination. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	86
3.32	<b>Small multiple representation of activity-policy matrices.</b> Each row represents all hourly activity-policy matrices for a given region in a small multiple setting to get an overview about temporal developments of network health. . . . .	87
3.33	<b>ClockMap visualization in BANKSAFE.</b> The different colored circles represent local computers establishing connections to IRC servers. The colored segments of each circle represent the number of connections over time. <i>Reprinted from [90]. © 2013 The authors.</i> . . . . .	89
3.34	<b>Degraded network activity after DoS attack.</b> The visualization shows network traffic on 2013-04-02, in which (A) represents 172.30.0.4 (WEB03.BIGMKT3.COM). Extreme peaks between 05:00 and 07:00 are caused by an ongoing DoS. However, the network traffic suddenly decreases between 07:00 and 08:00, and stays on a very low level (light yellow) which are are symptoms for major server issues (Event 6a). . . . .	91
3.35	<b>Detailed temporal network activity for main webserver.</b> High amount of network traffic originating from 172.30.0.4 until 07:02. No response until 07:16. Afterwards the web server recovers and resumes operations (Event 6b). . . . .	92
3.36	<b>Visualization for server crash on 2013-04-03.</b> (A) represents the host 172.30.0.4 (WEB03.BIGMKT3.COM) having lots of network activity followed by a long period without network traffic, indicating a major server crash (Event 9a). Other hosts in the subnet obviously do not experience such outages. . . . .	92

3.37	<b>Visualization of server return on 2013-04-05.</b> (A) represents the host 172.30.0.4 (WEB03.BIGMKT3.COM), which was offline after a server crash and returns back to operation between 07:00 and 08:00. Please note, that compared to Figure 3.36 the host’s location within the visualization has changed, because of the heavily reduced overall network traffic. . . . .	93
3.38	<b>Detection of subtle port scans.</b> The visualization of distinct ports on 2013-04-02. Using the normalization slider (A), subtle port scans from 10.7.7.10 (B) and 10.6.6.6 (C) between 13:00 and 14:00 can be identified (Event 7), which are not part of the more obvious DoS attack (Event 5) between 05:00 and 07:00. . . . .	94
3.39	<b>Multiple attackers conducting orchestrated port scans.</b> Various port scans (Event 17) from multiple attackers on 2013-04-10 are clearly visible. Using the normalization slider (A), the hosts stand out revealing quite distinctive patterns than related hosts in the respective subnets. . .	95
3.40	<b>Identification of port scans over longer periods of time.</b> The high peaks between 11:00 and 13:00 on 2013-04-11 are symptoms of a DoS attack (Event 19). Additionally, various port scans from attacker 10.12.15.152 (A) and 10.6.6.7 (B) over multiple hours relate to Event 18 and 21. . . . .	96
3.41	<b>Port scans of attackers belonging to the same subnet.</b> Port scans from 10.12.15.152 and 10.12.14.15 between 2013-04-12 11:00 and 16:00 (Event 24). . . . .	96
3.42	<b>Port scans of attackers originating from different subnets.</b> Port scans from 10.17.15.10 and 10.12.15.152 stick out with mostly red segments (high number of distinct utilized ports) starting at about 2013-04-13 05:00 (Event 25). . . . .	97
3.43	<b>Root cause identification of high network activity.</b> Exfiltration (Event 11) becomes visible on 2013-04-06 between 10:00 and 11:00. High network traffic in the subnet 172.10.0.0/24 (expanded in the highlighted rectangle) originates from a single host, which is the administrator’s workstation. Further exploration of the underlying network flow records reveals a data exfiltration to 10.7.5.5 of 109.6 MB via file transfer protocol (FTP). . . . .	97
3.44	<b>Identification of large data exfiltrations.</b> Another exfiltration becomes visible on 2013-04-07 between 07:00 and 08:00 using <i>ClockMap</i> (Event 14). The high traffic in the subnet 172.10.0.0/24 originates the administrator’s workstation (A). Further exploration of the underlying network flow records reveals another data exfiltration to 10.7.5.5 (about 650 MB). . . . .	98
3.45	<b>Identification of outliers in various subnets.</b> Compromised hosts become visible after a successful malware infection (Event 22). In subnet 172.30.1.0/24 two hosts stand out with their pattern (A). In subnet 172.20.1.0/24 three hosts (B) are visible very prominently, and another three hosts (C) are in the focus of 172.10.2.0/24. . . . .	98
3.46	<b>Distributed DoS by internal hosts on 2013-04-13.</b> Eight internal hosts start conducting an attack targeting an external webserver (Event 26). . . . .	99
3.47	<b>DoS attack originated by internal hosts on 2013-04-14.</b> The hosts continue to attack another external target on 2013-04-14 (Event 27). . . . .	100

3.48	<b>Integration of <i>ClockMap</i> into <i>VACS</i>.</b> The integration combines the strengths of all visualization techniques into a single web-based visual analytics suite. . . . .	101
------	--	-----

## Chapter 4

4.1	<b>Temporal MDS plots applied to network traffic data.</b> For each temporal MDS plot (top) the sequentially aligned matrix (bottom) provides an overview of correlations among dimensions. The visualization reveals the attack patterns for a distributed brute-force attack (A, D) and various different port scans (B, C). <i>Reprinted from [133]. © 2016 IEEE.</i> . . . . .	108
4.2	<b>Pixel-based visualization of <i>ELISHA</i>.</b> The main visualization consists of a scalable pixel-based approach to display BGP data. Each pixel represents an IP address with a color encoding according to the corresponding BGP event. The three detailed windows at the top enlarge areas of interest to better analyze single IP addresses. <i>Reprinted from [25]. © 2012 IEEE.</i> . . . . .	112
4.3	<b>Graphical user interface of <i>VisTracer</i>.</b> (1) and (2) provide access to constraint filters and a table with observed anomalies. (3) Visual ASN Overview with occurred anomalies. A Feedback Panel is provided in (4) and access to individual traceroutes in (5) with map-based (6), glyph-based (7) and graph-based (8) visualizations. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	117
4.4	<b>Visual analysis workflow in <i>VisTracer</i>.</b> The figure shows the overall interactive analysis workflow relating the various steps to the visualizations and views integrated in <i>VisTracer</i> . . . . .	120
4.5	<b>Suspicious AS networks.</b> Closeup of the <i>Visual ASN Overview</i> showing two nearly identical anomaly distributions for two different ASN at the same point in time. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	122
4.6	<b>Target History Visualization of the first case study.</b> The visualization shows the significant difference in the ASes traversed between the third and fourth day. The routing anomalies observed on the fourth day are also shown. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	122
4.7	<b>Graph visualization in <i>VisTracer</i>.</b> The node-link diagram with embedded clock glyphs shows significant difference in the ASes traversed between the third and fourth day. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	123
4.8	<b>Visual ASN Overview of AS31733.</b> The glyphs reveal many different anomalies over a longer period of time. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	124
4.9	<b>Target History Visualization of multiple traceroutes.</b> The figure shows the significant difference in the <i>set</i> of ASes traversed between the fourth and fifth day. The routing anomalies observed are also shown. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	125
4.10	<b>Temporal Graph Representation of the confirmed BGP hijack.</b> The figure shows the significant difference in the <i>sequence</i> of ASes traversed. It also highlights the unreachability of the destination AS after the routing change occurred. <i>Reprinted from [84]. © 2012 ACM.</i> . . . . .	126

4.11	<b>A Taxonomy of visual methods for malware analysis.</b> – Categorization of malware visualization systems into three categories, namely (1) Individual Malware Analysis, (2) Malware Comparison, and (3) Malware Summarization. All systems have the ultimate goal to generate rules and signatures for fully-automated malware detection systems. While the first category tackles the problem of understanding the behavior of an individual malware sample for forensics, the latter two focus on the identification of common behavior for malware classification. . . . .	130
4.12	<b>Comparison of malware images.</b> Visualizing malware executables as grayscale images is a common technique to visually identify similarities with low computation costs. . . . .	132
4.13	<b>Using VACS for visual attack attribution.</b> After feature selection and analysis, the shown visualization display can be used to explore the MDC clusters. The small-multiple view at the top can be used to select MDCs. The <i>Treemap View (TV)</i> , the <i>Graph View (GV)</i> , and the <i>Chord View (CV)</i> show the respective MDC of a well-known scam campaign impersonating the company “Eskom Holdings” [126]. . . . .	138
4.14	<b>Investigation during the field experiment.</b> Example of an MDC found during the field study, attributed to a notable espionage campaign. Reprinted from [88]. © 2014 The Eurographics Association. . . . .	139

## Chapter 5

5.1	<b>Real-time visualization display.</b> A basic visual display to monitor incoming live streams as raw messages and plot extracted geographic locations to a map. Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM. . . . .	149
5.2	<b>System architecture of NStreamAware.</b> Various modern systems, including <i>Apache Spark</i> , <i>RabbitMQ</i> , <i>MongoDB</i> , and <i>ElasticSearch</i> , to provide the needed scalability for an interactive visual analytics application for big data use cases. . . . .	151
5.3	<b>Using NVisAware to visualize heterogeneous data streams.</b> The figure shows various real-time sliding slices. New slices of the most recent sliding window will automatically added on the right. . . . .	154
5.4	<b>Using visual feature selection for NVisAware.</b> All temporal features can be visualized as time-series for feature selection. . . . .	155
5.5	<b>Example of selected features.</b> The analyst is in the loop to steer the merging algorithm through selecting, ranking, and modifying feature segments, to provide meaningful summaries of sliding slices. Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM. . . . .	156
5.6	<b>Overview of the interactive visual analytics workflow.</b> Visualization with <i>NVisAware</i> is used for real-time analysis, but also for summary visualizations based on the multi-focal aggregation of sliding slices using an interactive visual feature selection process. . . . .	157

5.7	<b>Example of findings using <i>NVisAware</i>.</b> Visualization to monitor data streams using sliding slices to reveal interesting findings. The interactive display can be explored by the analyst while new sliding slices are continuously added. <i>Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	158
5.8	<b>Example of summarization results.</b> Visual feature selection helps to merge many slices to single summary slices. <i>Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	159
5.9	<b>Timeline of major VAST Challenge 2014 MC3 events.</b> The colored histogram highlights major events based on extracted keywords and insights of interesting events, which could be identified in real-time. <i>Reprinted from [79]. © 2014 Copyright is held by the owner/author(s). Publication rights licensed to ACM.</i> . . . . .	160
5.10	<b>Using <i>NVisAware</i> to solve the VAST Challenge 2014 MC3.</b> New sliding slices are automatically added on the right to summarize the most recent events happening on Kronos island. . . . .	161
5.11	<b>An example for a major fire incident.</b> Three stream slices out of many slices of the data stream to show the evolving topics in the word cloud. The red background color reflects a high word cloud dissimilarity to the previous slices. . . . .	162

# List of Tables

## Chapter 2

2.1	<b>State-of-the-art overview according to paper type.</b> The table gives an overview of the general paper types included in the overall literature review. . . . .	14
2.2	<b>State-of-the-art overview of related surveys.</b> Categorization of papers reviewed by various existing surveys. Some of the papers in this literature review were also discussed in previously published surveys. This table gives an overview, which papers have been reviewed in the respective surveys. . . . .	15
2.3	<b>Overview of yearly trends for situational awareness.</b> The table gives an overview about the number of methods with respect to stages for situational awareness, uses of visualization, types of analysis, and use cases. Only few visualization systems address the projection stage or focus on the communication of insights. Threat analysis and attack attribution use cases are also underrepresented in research. . . . .	19
2.4	<b>State-of-the-art overview based on primary use case.</b> The adapted and extended use case classification based on Shiravi et al. [216] helps to group the approaches into various distinctive general use cases. Each approach is assigned to a single use case category, which represents the primary use case respectively. . . . .	20
2.5	<b>State-of-the-art overview based on primary data sources.</b> This overview represents the primarily used data sources in the reviewed methods. . . . .	28
2.6	<b>Yearly trends for visualization types and techniques.</b> The table gives an overview about the most widely used visualization types based on a general taxonomy by Keim [136] and various common visualization techniques. . . . .	29
2.7	<b>Yearly trends for used evaluation techniques.</b> An overview about the most widely used evaluation techniques in the reviewed methods and applications. Obviously, case studies and usage scenarios are the most widely used technique, which we categorize as <i>insight-based strategies</i> . . . . .	30

## Chapter 3

3.1	<b>Related work for internal/external monitoring.</b> Overview of related work with respect to data source and visualization type. . . . .	41
3.2	<b>Evaluation of VACS using VAST Challenge 2013 MC3.</b> . . . . .	51
3.3	<b>Related work for port activity monitoring.</b> Overview of related work with respect to data source and visualization type. . . . .	60
3.4	<b>Overview of selected descriptors.</b> A time-series group containing some of the network time-series belonging to the most widely exploited services. . . . .	65
3.5	<b>Overview of automatically retrieved descriptors.</b> The first seven series returned by the server when the analyst queried for the anomaly region (A) he visually identified as seen in Figure 3.19. . . . .	67

3.6	<b>Related work with methods for host and server monitoring.</b>	
	Overview of related work with respect to data source and visualization type.	70
3.7	Overview of compared glyphs within the user study. . . . .	78
3.8	<b>Summary of reviewer scores for <i>BANKSAFE</i>.</b> The table presents the scores given by the anonymous reviewers (R1-7) for our VAST Challenge 2012 solution with submission ID #118 focusing on MC2 featuring our <i>ClockMap</i> approach. . . . .	89
3.9	<b>Ground truth evaluation for <i>ClockMap</i>.</b> . . . . .	90

## Chapter 4

4.1	<b>Related work for attack pattern visualization methods.</b> Overview of related work with respect to data source and visualization type. . . . .	105
4.2	<b>Related work of visualization methods for routing behavior.</b> Overview of related work for visual analysis of routing behavior and anomaly detection. . . . .	111
4.3	<b>Various glyphs used in <i>VisTracer</i> visualizations.</b> An overview about the three glyphs, which are incorporated in the various visualizations in <i>VisTracer</i> . . . . .	118
4.4	<b>State-of-the-art overview for visual malware analysis.</b> Overview of state-of-the-art techniques for visual analysis of malware behavior. . . . .	128
4.5	<b>Related work for analyzing the threat landscape.</b> Overview of related work to analyze the threat landscape and provide visual support for attack attribution. . . . .	136
4.6	<b>High-level summary of qualitative feedback.</b> A selection of results based on qualitative feedback during the field experiment. . . . .	142

## Chapter 5

5.1	<b>Features for various network-related data streams.</b> The table shows a selection of aggregation features, which are automatically calculated in regular intervals by our implemented analysis and aggregation module for each sliding window. . . . .	153
-----	--	-----



## Bibliography

- [1] K. Abdullah, C. Lee, G. Conti, and J. Copeland. Visualizing network data for intrusion detection. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, pages 100–108, 2005. doi:[10.1109/IAW.2005.1495940](https://doi.org/10.1109/IAW.2005.1495940). [pages 15, 20, 28, and 60]
- [2] K. Abdullah, C. Lee, G. Conti, J. Copeland, and J. Stasko. IDS RainStorm: Visualizing IDS Alarms. *IEEE Workshop on Visualization for Computer Security*, pages 1–10, 2005. doi:[10.1109/VIZSEC.2005.1532060](https://doi.org/10.1109/VIZSEC.2005.1532060). [pages 15, 20, 28, and 105]
- [3] E. Adam. Fighter cockpits of the future. In *AIAA/IEEE Digital Avionics Systems Conference, 1993. 12th DASC*, pages 318–323, 1993. doi:[10.1109/DASC.1993.283529](https://doi.org/10.1109/DASC.1993.283529). [page 9]
- [4] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A Framework for Clustering Evolving Data Streams. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03*, pages 81–92, Berlin, Germany, 2003. VLDB Endowment. ISBN 0-12-722442-4. [page 148]
- [5] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction. Springer Verlag, London, UK, 1st edition, 2011. ISBN 978-0-85729-078-6. [page 59]
- [6] C. Alexander. *The Nature of Order: An Essay on the Art of Building and the Nature of the Universe: Book I - The Phenomenon of Life*. The Center for Environmental Structure, Berkeley, CA, USA, 2002. [pages 32 and 81]
- [7] M. Alsaleh, A. Alqahtani, A. Alarifi, and A. Al-Salman. Visualizing PHPIDS Log Files for Better Understanding of Web Server Attacks. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security, VizSec '13*, pages 1–8, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:[10.1145/2517957.2517958](https://doi.org/10.1145/2517957.2517958). [pages 14, 20, 28, 104, and 105]
- [8] Z. Alshaikh, A. Alarifi, and M. Alsaleh. Christopher Alexander’s fifteen properties: Toward developing evaluation metrics for security visualizations. In *2013 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 295–300, 2013. doi:[10.1109/ISI.2013.6578847](https://doi.org/10.1109/ISI.2013.6578847). [pages 14, 32, 77, 80, and 81]
- [9] B. Anderson, C. Storlie, and T. Lane. Improving Malware Classification: Bridging the Static/Dynamic Gap. In *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence, AISEC '12*, pages 3–14, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1664-4. doi:[10.1145/2381896.2381900](https://doi.org/10.1145/2381896.2381900). [pages 14, 15, 20, 28, 128, 130, and 133]
- [10] D. F. Andrews. Plots of High-Dimensional Data. *Biometrics*, 28(1):125–136, 1972. ISSN 0006-341X. doi:[10.2307/2528964](https://doi.org/10.2307/2528964). [page 27]
- [11] Apache. Spark Streaming. URL <https://spark.apache.org/streaming/>. Accessed: 2015-10-01. [pages 146 and 150]

- [12] D. Arendt, R. Burtner, D. Best, N. Bos, J. Gersh, C. Piatko, and C. Paul. Ocelot: user-centered design of a decision support visualization for network quarantine. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, 2015. doi:[10.1109/VIZSEC.2015.7312763](https://doi.org/10.1109/VIZSEC.2015.7312763). [pages 14, 20, 28, and 70]
- [13] R. Arias-Hernandez, L. Kaastra, T. Green, and B. Fisher. Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics. In *2011 44th Hawaii International Conference on System Sciences (HICSS)*, pages 1–10, 2011. doi:[10.1109/HICSS.2011.339](https://doi.org/10.1109/HICSS.2011.339). [page 31]
- [14] R. Ball, G. A. Fink, and C. North. Home-centric Visualization of Network Traffic for Security Administration. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04*, pages 55–64, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029217](https://doi.org/10.1145/1029208.1029217). [pages 14, 15, 19, 20, 28, 40, and 41]
- [15] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the Internet. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '07*, pages 265–276, Kyoto, Japan, 2007. ACM. ISBN 978-1-59593-713-1. doi:[10.1145/1282380.1282411](https://doi.org/10.1145/1282380.1282411). [page 113]
- [16] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi Treemaps for the Visualization of Software Metrics. In *Proceedings of the 2005 ACM symposium on Software visualization, SoftVis '05*, pages 165–172, St. Louis, Missouri, 2005. ACM. ISBN 1-59593-073-6. [page 71]
- [17] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia, S. Jha, J. Li, P. Liu, P. Ning, X. Ou, D. Song, L. Strater, V. Swarup, G. Tadda, C. Wang, and J. Yen. Cyber SA: Situational Awareness for Cyber Defense. In S. Jajodia, P. Liu, V. Swarup, and C. Wang, editors, *Cyber Situational Awareness*, number 46 in Advances in Information Security, pages 3–13. Springer US, 2010. ISBN 978-1-4419-0139-2, 978-1-4419-0140-8. [page 10]
- [18] G. D. Battista, F. Mariani, M. Patrignani, and M. Pizzonia. BGPlay: A System for Visualizing the Interdomain Routing Evolution. In G. Liotta, editor, *Graph Drawing*, number 2912 in Lecture Notes in Computer Science, pages 295–306. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-20831-0 978-3-540-24595-7. [pages 14, 15, 20, 28, 36, 110, 111, and 113]
- [19] J. M. Beaver, C. A. Steed, R. M. Patton, X. Cui, and M. Schultz. Visualization techniques for computer network defense. volume 8019, pages 801906–801906–9, 2011. doi:[10.1117/12.883487](https://doi.org/10.1117/12.883487). [pages 15, 20, 28, and 70]
- [20] M. Behrisch, J. Davey, F. Fischer, O. Thonnard, T. Schreck, D. Keim, and J. Kohlhammer. Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom. *Computer Graphics Forum*, 33(3):411–420, 2014. ISSN 1467-8659. doi:[10.1111/cgf.12397](https://doi.org/10.1111/cgf.12397). [pages 8 and 169]
- [21] R. Berthier, M. Cukier, M. Hiltunen, D. Kormann, G. Vesonder, and D. Sheheda. Nfsight: netflow-based network awareness tool. In *Proceedings of the 24th*

- international conference on Large installation system administration*, pages 1–8. USENIX Association, 2010. [pages 14, 20, 28, and 70]
- [22] E. Bertini, P. Hertzog, and D. Lalanne. SpiralView: Towards Security Policies Assessment through Visual Correlation of Network Resources with Evolution of Alarms. *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 139–146, 2007. doi:[10.1109/VAST.2007.4389007](https://doi.org/10.1109/VAST.2007.4389007). [pages 14, 15, 20, 28, and 105]
- [23] E. Bertini, J. Buchmüller, F. Fischer, S. Huber, T. Lindemeier, F. Maaß, F. Mansmann, T. Ramm, M. Regenscheit, C. Rohrdantz, C. Scheible, T. Schreck, S. Sellien, F. Stoffel, M. Tautzenberger, M. Zieker, and D. A. Keim. Visual Analytics of Terrorist Activities Related to Epidemics. In *Proc. IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2011 - Grand Challenge Award)*, 2011. doi:[10.1109/VAST.2011.6102498](https://doi.org/10.1109/VAST.2011.6102498). [page 7]
- [24] D. M. Best, S. Bohn, D. Love, A. Wynne, and W. A. Pike. Real-Time Visualization of Network Behaviors for Situational Awareness. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security, VizSec '10*, pages 79–90, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0013-1. doi:[10.1145/1850795.1850805](https://doi.org/10.1145/1850795.1850805). [pages 14, 15, 20, 28, 60, 70, and 148]
- [25] E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P.-A. Vervier. Visual Analytics for BGP Monitoring and Prefix Hijacking Identification. *IEEE Network*, 26(6):33–39, 2012. ISSN 0890-8044. doi:[10.1109/MNET.2012.6375891](https://doi.org/10.1109/MNET.2012.6375891). [pages 4, 11, 13, 14, 15, 20, 28, 110, 112, and 178]
- [26] P. Bloomfield. *Fourier Analysis of Time Series: An Introduction*. John Wiley & Sons, 2004. ISBN 978-0-471-65399-8. [page 61]
- [27] R. Blue, C. Dunne, A. Fuchs, K. King, and A. Schulman. Visualizing Real-Time Network Resource Usage. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, number 5210 in Lecture Notes in Computer Science, pages 119–135. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4 978-3-540-85933-8. [pages 14, 20, 28, 36, and 41]
- [28] A. Boschetti, L. Salgarelli, C. Muelder, and K.-L. Ma. TVi: A Visual Querying System for Network Monitoring and Anomaly Detection. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11*, pages 1:1–1:10, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0679-9. doi:[10.1145/2016904.2016905](https://doi.org/10.1145/2016904.2016905). [pages 14, 15, 20, 28, 41, and 107]
- [29] M. Bostock, V. Ogievetsky, and J. Heer. D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. ISSN 1077-2626. doi:[10.1109/TVCG.2011.185](https://doi.org/10.1109/TVCG.2011.185). [pages 83 and 138]
- [30] J. M. Bradshaw, M. Carvalho, L. Bunch, T. Eskridge, P. J. Feltovich, M. Johnson, and D. Kidwell. Sol: An Agent-Based Framework for Cyber Situation Awareness. *KI - Künstliche Intelligenz*, 26(2):127–140, 2012. ISSN 0933-1875, 1610-1987. doi:[10.1007/s13218-012-0179-2](https://doi.org/10.1007/s13218-012-0179-2). [pages 14, 20, 28, and 105]

- [31] C. A. Brewer. ColorBrewer 2.0 - Color Advice for Cartography. URL <http://colorbrewer2.org/>. Accessed: 2015-10-01. [pages 72, 118, and 153]
- [32] M. Bruls, K. Huizing, and J. J. v. Wijk. Squarified Treemaps. In D. i. W. C. d. Leeuw and i. R. v. Liere, editors, *Data Visualization 2000*, Eurographics, pages 33–42. Springer Vienna, 2000. ISBN 978-3-211-83515-9 978-3-7091-6783-0. [page 137]
- [33] BSI. Die Lage der IT-Sicherheit in Deutschland 2015 (in German). URL [http://docs.dpaq.de/9977-2015\\_11\\_19\\_bsi\\_lagebericht\\_2015.pdf](http://docs.dpaq.de/9977-2015_11_19_bsi_lagebericht_2015.pdf). Accessed: 2015-12-01. [page 1]
- [34] J. Buchmüller, F. Fischer, D. Streeb, and D. A. Keim. Using visual analytics to provide situation awareness for movement and communication data. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–122, 2015. doi:10.1109/VAST.2015.7347640. [page 8]
- [35] R. Bush and R. Austein. The RPKI & Origin Validation. URL <http://meetings.internet2.edu/2009-07-JT/detail/10000724/>. Accessed: 2015-10-01. [page 113]
- [36] E. Cakmak, A. Gartner, T. Hepp, J. Buchmüller, F. Fischer, and D. A. Keim. Applying visual analytics to explore and analyze movement data. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 127–128, 2015. doi:10.1109/VAST.2015.7347643. [page 8]
- [37] B. Cappers and J. van Wijk. SNAPS: Semantic network traffic analysis through projection and selection. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, 2015. doi:10.1109/VIZSEC.2015.7312768. [pages 14, 20, 28, 104, and 105]
- [38] S. Carpendale. Evaluating Information Visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization*, number 4950 in Lecture Notes in Computer Science, pages 19–45. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-70955-8 978-3-540-70956-5. [pages 30, 31, 32, and 139]
- [39] S. Carpendale, J. Ligh, and E. Pattison. Achieving Higher Magnification in Context. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 71–80, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi:10.1145/1029632.1029645. [page 64]
- [40] N. Cawthon and A. Moere. The Effect of Aesthetic on the Usability of Data Visualization. In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 637–648, 2007. doi:10.1109/IV.2007.147. [page 80]
- [41] S. Chen, C. Guo, X. Yuan, F. Merkle, H. Schaefer, and T. Ertl. OCEANS: Online Collaborative Explorative Analysis on Network Security. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, VizSec '14, pages 1–8, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671493. [pages 14, 20, 28, and 105]
- [42] V. Y. Chen, A. M. Razip, S. Ko, C. Z. Qian, and D. S. Ebert. Multi-aspect visual analytics on large-scale high-dimensional cyber security data.

- Information Visualization*, 14(1):62–75, 2015. ISSN 1473-8716, 1473-8724. doi:[10.1177/1473871613488573](https://doi.org/10.1177/1473871613488573). [pages 14, 20, 28, and 70]
- [43] H. Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973. ISSN 0162-1459. doi:[10.1080/01621459.1973.10482434](https://doi.org/10.1080/01621459.1973.10482434). [page 27]
- [44] G. Chin, M. Singhal, G. Nakamura, V. Gurumoorthi, and N. Freeman-Cadoret. Visual analysis of dynamic data streams. *Information Visualization*, 8(3):212–229, 2009. ISSN 1473-8716. doi:[10.1057/ivs.2009.18](https://doi.org/10.1057/ivs.2009.18). [page 71]
- [45] H. Choi and H. Lee. PCAV: Internet Attack Visualization on Parallel Coordinates. In S. Qing, W. Mao, J. López, and G. Wang, editors, *Information and Communications Security*, number 3783 in Lecture Notes in Computer Science, pages 454–466. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-30934-5 978-3-540-32099-9. [pages 14, 15, 20, 28, and 105]
- [46] H. Choi, H. Lee, and H. Kim. Fast detection and visualization of network attacks on parallel coordinates. *Computers & Security*, 28(5):276–288, 2009. ISSN 0167-4048. doi:[10.1016/j.cose.2008.12.003](https://doi.org/10.1016/j.cose.2008.12.003). [pages 14, 15, 20, 28, and 105]
- [47] M. Chu, K. Ingols, R. Lippmann, S. Webster, and S. Boyer. Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*, pages 22–33. ACM, 2010. doi:[10.1145/1850795.1850798](https://doi.org/10.1145/1850795.1850798). [pages 14, 20, 28, 104, and 105]
- [48] W. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, pages 531–554, 1984. [page 78]
- [49] L. Colitti, G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Visualizing Interdomain Routing with BGPlay. *Journal of Graph Algorithms and Applications*, 9(1):117–148, 2005. [pages 14, 15, 20, 28, and 111]
- [50] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. A. Copeland, M. Ahamad, H. L. Owen, and C. Lee. Countering Security Information Overload Through Alert and Packet Visualization. *IEEE Comput. Graph. Appl.*, 26(2):60–70, 2006. ISSN 0272-1716. doi:[10.1109/MCG.2006.30](https://doi.org/10.1109/MCG.2006.30). [pages 14, 15, 20, 28, 104, and 105]
- [51] G. Conti, E. Dean, M. Sinda, and B. Sangster. Visual Reverse Engineering of Binary and Data Files. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, number 5210 in Lecture Notes in Computer Science, pages 1–17. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4, 978-3-540-85933-8. [pages 14, 15, 20, 28, 128, and 130]
- [52] K. Cook, G. Grinstein, M. Whiting, M. Cooper, P. Havig, K. Liggett, B. Nebesh, and C. L. Paul. VAST Challenge 2012: Visual Analytics for Big Data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 251–255, 2012. doi:[10.1109/VAST.2012.6400529](https://doi.org/10.1109/VAST.2012.6400529). [pages 6, 82, 83, and 88]
- [53] E. Corchado and A. Herrero. Neural Visualization of Network Traffic Data for Intrusion Detection. *Appl. Soft Comput.*, 11(2):2042–2056, 2011. ISSN 1568-4946. doi:[10.1016/j.asoc.2010.07.002](https://doi.org/10.1016/j.asoc.2010.07.002). [pages 14, 15, 20, 28, and 105]

- [54] I. Corp. RFC3176 - InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. URL <https://tools.ietf.org/html/rfc3176>. Accessed: 2015-10-01. [page 41]
- [55] P. F. Cortese, G. Di Battista, A. Moneta, M. Patrignani, and M. Pizzonia. Topographic visualization of prefix propagation in the internet. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):725–32, 2006. ISSN 1077-2626. doi:10.1109/TVCG.2006.185. [pages 14, 15, 20, 28, and 111]
- [56] M. Cova, C. Leita, O. Thonnard, A. D. Keromytis, and M. Dacier. An Analysis of Rogue AV Campaigns. In S. Jha, R. Sommer, and C. Kreibich, editors, *Recent Advances in Intrusion Detection*, number 6307 in Lecture Notes in Computer Science, pages 442–463. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15511-6 978-3-642-15512-3. [page 137]
- [57] M. Dacier, V. Pham, and O. Thonnard. The WOMBAT Attack Attribution method: some results. *Information Systems Security*, pages 19–37, 2009. [pages 22 and 135]
- [58] A. D'Amico and M. Kocka. Information assurance visualizations for specific stages of situational awareness and intended uses: lessons learned. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*, pages 107–112, 2005. doi:10.1109/VIZSEC.2005.1532072. [pages 15, 16, 17, 18, and 173]
- [59] DARPA. RFC791 - Internet Protocol Specification, . URL <https://tools.ietf.org/html/rfc791>. Accessed: 2015-10-01. [page 41]
- [60] DARPA. RFC793 - Transmission Control Protocol, . URL <https://tools.ietf.org/html/rfc793>. Accessed: 2015-10-01. [page 41]
- [61] David A. Wheeler and Gregory N. Larsen. Techniques for Cyber Attack Attribution. *Institute for Defense Analyses*, page 82, 2003. [page 22]
- [62] S. Diehl, F. Beck, and M. Burch. Uncovering Strengths and Weaknesses of Radial Visualizations—an Empirical Approach. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):935–942, 2010. ISSN 1077-2626. doi:10.1109/TVCG.2010.209. [page 76]
- [63] J. Donahue, A. Paturi, and S. Mukkamala. Visualization techniques for efficient malware detection. In *2013 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 289–291, 2013. doi:10.1109/ISI.2013.6578845. [pages 14, 15, 20, 28, 128, and 130]
- [64] M. El Assady, W. Jentner, M. Stein, F. Fischer, T. Schreck, and D. A. Keim. Predictive Visual Analytics – Approaches for Movie Ratings and Discussion of Open Research Challenges. In *Proceedings of the IEEE VIS 2014 Workshop Visualization for Predictive Analytics*, 2014. [page 8]
- [65] J. Ellson, E. Gansner, L. Koutsofios, S. North, and G. Woodhull. Graphviz—Open Source Graph Drawing Tools. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*, pages 483–484. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-43309-5. [page 138]

- [66] M. R. Endsley. Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 32 (2):97–101, 1988. ISSN 1541-9312,. doi:[10.1177/154193128803200221](https://doi.org/10.1177/154193128803200221). [page 9]
- [67] M. R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37 (1):32–64, 1995. doi:[10.1518/001872095779049543](https://doi.org/10.1518/001872095779049543). [pages 9, 10, 15, and 148]
- [68] R. Erbacher. Intrusion Behavior Detection Through Visualization. In *IEEE International Conference on Systems, Man and Cybernetics, 2003*, volume 3, pages 2507–2513 vol.3, 2003. doi:[10.1109/ICSMC.2003.1244260](https://doi.org/10.1109/ICSMC.2003.1244260). [pages 14, 15, 20, 28, and 70]
- [69] R. Erbacher, K. Walker, and D. Frincke. Intrusion and Misuse Detection in Large-Scale Systems. *IEEE Computer Graphics and Applications*, 22(1):38–47, 2002. ISSN 0272-1716. doi:[10.1109/38.974517](https://doi.org/10.1109/38.974517). [pages 14, 15, 20, 28, and 70]
- [70] R. Erbacher, K. Christensen, and A. Sundberg. Designing visualization capabilities for IDS challenges. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*, pages 121–127, 2005. doi:[10.1109/VIZSEC.2005.1532074](https://doi.org/10.1109/VIZSEC.2005.1532074). [pages 14, 15, 20, 28, and 41]
- [71] R. F. Erbacher. Visualization Design for Immediate High-level Situational Assessment. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 17–24, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:[10.1145/2379690.2379693](https://doi.org/10.1145/2379690.2379693). [pages 14, 15, 20, 28, 70, and 148]
- [72] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996. [page 155]
- [73] M. Eto, D. Inoue, J. Song, J. Nakazato, K. Ohtaka, and K. Nakao. Nicter: A Large-scale Network Incident Analysis System: Case Studies for Understanding Threat Landscape. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS '11*, pages 37–45, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0768-0. doi:[10.1145/1978672.1978677](https://doi.org/10.1145/1978672.1978677). [pages 14, 20, 28, and 136]
- [74] S. Few. Bullet graph design specification. *Perceptual Edge-White Paper*, 2013. [page 43]
- [75] S. Few. *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Analytics Press, Burlingame, Calif., second edition, second edition edition, 2013. ISBN 978-1-938377-00-6. [page 43]
- [76] G. Fink, C. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009*, pages 45–56, 2009. doi:[10.1109/VIZSEC.2009.5375542](https://doi.org/10.1109/VIZSEC.2009.5375542). [page 59]

- [77] G. A. Fink, P. Muessig, and C. North. Visual Correlation of Host Processes and Network Traffic. In *Visualization for Computer Security, IEEE Workshops on*, volume 0, page 2, Los Alamitos, CA, USA, 2005. IEEE Computer Society. ISBN 0-7803-9477-1. doi:[10.1109/VIZSEC.2005.18](https://doi.org/10.1109/VIZSEC.2005.18). [pages 15, 20, 28, and 70]
- [78] F. Fischer and D. A. Keim. VACS: Visual Analytics Suite for Cyber Security - Visual Exploration of Cyber Security Datasets. In *VAST Challenge 2013 - Honorable Mention*, 2013. [pages 6, 14, 20, 28, and 42]
- [79] F. Fischer and D. A. Keim. NStreamAware: Real-Time Visual Analytics for Data Streams to Enhance Situational Awareness. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 65–72, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:[10.1145/2671491.2671495](https://doi.org/10.1145/2671491.2671495). [pages 5, 14, 20, 28, 146, 149, 156, 158, 159, 160, 179, and 180]
- [80] F. Fischer and F. Stoffel. NStreamAware: Real-Time Visual Analytics for Data Streams (VAST Challenge 2014 MC3). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 373–374, 2014. doi:[10.1109/VAST.2014.7042572](https://doi.org/10.1109/VAST.2014.7042572). [pages 6 and 146]
- [81] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-Scale Network Monitoring for Visual Analysis of Attacks. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, number 5210 in Lecture Notes in Computer Science, pages 111–118. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4, 978-3-540-85933-8. [pages 14, 15, 20, 21, 28, 104, 105, 107, 137, and 173]
- [82] F. Fischer, J. Fuchs, and F. Mansmann. ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data. In M. Meyer and T. Weinkauff, editors, *Proceedings of the Eurographics Conference on Visualization (EuroVis - Short Papers)*, pages 97–101, Vienna, Austria, 2012. The Eurographics Association. ISBN 978-3-905673-91-3. doi:[10.2312/PE/EuroVisShort/EuroVisShort2012/097-101](https://doi.org/10.2312/PE/EuroVisShort/EuroVisShort2012/097-101). [pages 5, 14, 15, 20, 28, 52, 69, 72, 73, 81, and 175]
- [83] F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: A Visual Situational Awareness Tool for Large-Scale Computer Networks (VAST Challenge 2012). In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 257–258, 2012. doi:[10.1109/VAST.2012.6400528](https://doi.org/10.1109/VAST.2012.6400528). [pages 6 and 82]
- [84] F. Fischer, J. Fuchs, P.-A. Vervier, F. Mansmann, and O. Thonnard. VisTracer: A Visual Analytics Tool to Investigate Routing Anomalies in Traceroutes. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 80–87, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:[10.1145/2379690.2379701](https://doi.org/10.1145/2379690.2379701). [pages 5, 14, 20, 28, 110, 117, 122, 123, 124, 125, 126, and 178]
- [85] F. Fischer, F. Mansmann, and D. A. Keim. Real-Time Visual Analytics for Event Data Streams. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*, pages 801–806, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0857-1. doi:[10.1145/2245276.2245432](https://doi.org/10.1145/2245276.2245432). [pages 14, 20, 28, 70, 87, and 148]



- [86] F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. Visual Analytics zur Firewall-Konfiguration und Analyse von Netzwerkverkehr (in German). In B. f. S. i. d. Informationstechnik, editor, *Informationssicherheit stärken - Vertrauen in die Zukunft schaffen: Tagungsband zum 13. Deutschen IT-Sicherheitskongress (in German)*, pages 273–283. SecuMedia Verlag, 2013. [pages 5 and 69]
- [87] F. Fischer, D. Jäckle, D. Sacha, F. Stoffel, and D. A. Keim. Adaptive User-Aware Dashboard Design. In *VAST Challenge 2013 - Honorable Mention*, 2013. [pages 6, 17, 43, and 48]
- [88] F. Fischer, J. Davey, J. Fuchs, O. Thonnard, J. Kohlhammer, and D. A. Keim. A Visual Analytics Field Experiment to Evaluate Alternative Visualizations for Cyber Security Applications. In M. Pohl and J. Roberts, editors, *Proc. EuroVA International Workshop on Visual Analytics*. The Eurographics Association, 2014. ISBN 978-3-905674-68-2. doi:10.2312/eurova.20141144. [pages 6, 14, 20, 28, 45, 135, 139, and 179]
- [89] F. Fischer, F. Stoffel, S. Mittelstädt, T. Schreck, and D. A. Keim. Using Visual Analytics to Support Decision Making to Solve the Kronos Incident (VAST Challenge 2014). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 301–302, 2014. doi:10.1109/VAST.2014.7042537. [page 6]
- [90] F. Fischer, J. Fuchs, F. Mansmann, and D. A. Keim. BANKSAFE: Visual Analytics for Big Data in Large-Scale Computer Networks. *Information Visualization*, 14(1):51–61, 2015. ISSN 1473-8716, 1473-8724. doi:10.1177/1473871613488572. [pages 5, 14, 15, 20, 28, 82, 83, 84, 85, 86, 89, 167, and 176]
- [91] S. Foresti, J. Agutter, Y. Livnat, S. Moon, and R. Erbacher. Visual Correlation of Network Alerts. *IEEE Computer Graphics and Applications*, 26:48–59, 2006. ISSN 0272-1716. doi:10.1109/MCG.2006.49. [pages 14, 15, 20, 28, and 105]
- [92] J. J. Fowler, T. Johnson, P. Simonetto, M. Schneider, C. Acedo, S. Kobourov, and L. Lazos. IMap: Visualizing Network Activity over Internet Maps. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 80–87, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671501. [pages 14, 20, 28, 33, and 105]
- [93] U. Franke and J. Brynielsson. Cyber Situational Awareness - A Systematic Review of the Literature. *Computers & Security*, 46:18 – 31, 2014. ISSN 0167-4048. doi:10.1016/j.cose.2014.06.008. [pages 10, 11, 14, and 15]
- [94] A. Frei and M. Rennhard. Histogram Matrix: Log File Visualization for Anomaly Detection. *2008 Third International Conference on Availability, Reliability and Security*, pages 610–617, 2008. doi:10.1109/ARES.2008.148. [pages 14, 20, 28, and 70]
- [95] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pages 3237–3246, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi:10.1145/2470654.2466443. [pages 6, 77, 78, 79, 80, and 167]

- [96] J. Fuchs, P. Isenberg, A. Bezerianos, F. Fischer, and E. Bertini. The Influence of Contour on Similarity Perception of Star Glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2251–2260, 2014. ISSN 1077-2626. doi:10.1109/TVCG.2014.2346426. [page 8]
- [97] J. Fuchs, R. Rädle, D. Sacha, F. Fischer, and A. Stoffel. Collaborative Data Analysis with Smart Tangible Devices. In *Proceedings of Conference on Visualization and Data Analysis (VDA '14)*, volume 9017, pages 90170C–90170C–15, 2014. doi:10.1117/12.2040011. [page 8]
- [98] M. Ghoniem, G. Shurkhovetsky, A. Bahey, and B. Otjacques. VAFLE: Visual analytics of firewall log events. volume 9017, pages 901704–901704–15, 2013. doi:10.1117/12.2037790. [pages 14, 20, 28, and 70]
- [99] L. Girardin. An Eye on Network Intruder-administrator Shootouts. In *Proceedings of the 1st Conference on Workshop on Intrusion Detection and Network Monitoring - Volume 1*, ID'99, pages 3–3, Berkeley, CA, USA, 1999. USENIX Association. [pages 14, 15, 20, 28, and 105]
- [100] J. Goodall. Visualization is better! A comparative evaluation. In *6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009*, pages 57–68, 2009. doi:10.1109/VIZSEC.2009.5375543. [pages 2 and 14]
- [101] J. Goodall and M. Sowul. VIAssist: Visual analytics for cyber defense. In *IEEE Conference on Technologies for Homeland Security, 2009. HST '09*, pages 143–150, 2009. doi:10.1109/THS.2009.5168026. [pages 14, 20, 28, 36, and 105]
- [102] J. Goodall, W. Lutters, P. Rheingans, and A. Komlodi. Preserving the big picture: visual network traffic analysis with TNV. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*, pages 47–54, 2005. doi:10.1109/VIZSEC.2005.1532065. [pages 14, 15, 20, 28, 41, and 107]
- [103] Google. Google BigQuery. URL <http://developers.google.com/bigquery/>. Accessed: 2013-01-11. [page 83]
- [104] R. Gove, J. Saxe, S. Gold, A. Long, and G. Bergamo. SEEM: A Scalable Visualization for Comparing Multiple Large Sets of Attributes for Malware Analysis. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 72–79, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671496. [pages 14, 15, 20, 28, 128, 130, and 131]
- [105] A. R. A. Grégio and R. D. C. Santos. Visualization techniques for malware behavior analysis. In *Proc. Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X*, volume 8019 of *SPIE 8019*, pages 801905–801905–9, 2011. doi:10.1117/12.883441. [pages 14, 15, 20, 28, 128, and 130]
- [106] A. R. A. Grégio, A. O. C. Baruque, V. M. Afonso, D. S. F. Filho, P. L. d. Geus, M. Jino, and R. D. C. d. Santos. Interactive, Visual-Aided Tools to Analyze Malware Behavior. In B. Murgante, O. Gervasi, S. Misra, N. Nedjah, A. M. A. C. Rocha, D. Taniar, and B. O. Apduhan, editors, *Computational Science and Its Applications – ICCSA 2012*, number 7336 in *Lecture Notes in Computer*

- Science, pages 302–313. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-31127-7, 978-3-642-31128-4. [pages 14, 15, 20, 28, 128, 130, and 131]
- [107] N. W. Group. RFC2460 - Internet Protocol, Version 6 (IPV6) Specification. URL <https://tools.ietf.org/html/rfc2460>. Accessed: 2015-10-01. [page 41]
- [108] V. Guimaraes, C. Dal Sasso Freitas, R. Sadre, L. Tarouco, and L. Granville. A Survey on Information Visualization for Network and Service Management. *IEEE Communications Surveys Tutorials*, PP(99):1–1, 2015. ISSN 1553-877X. doi:10.1109/COMST.2015.2450538. [pages 11, 14, and 15]
- [109] K. Han, J. H. Lim, and E. G. Im. Malware Analysis Method Using Visualization of Binary Files. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, RACS '13, pages 317–321, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2348-2. doi:10.1145/2513228.2513294. [pages 14, 15, 20, 28, 128, 130, and 133]
- [110] K. Han, B. Kang, and E. G. Im. Malware Analysis Using Visualized Image Matrices. *The Scientific World Journal*, 2014:e132713, 2014. ISSN 2356-6140. doi:10.1155/2014/132713. [pages 14, 15, 20, 28, 128, 130, and 133]
- [111] K. S. Han, J. H. Lim, B. Kang, and E. G. Im. Malware analysis using visualized images and entropy graphs. *International Journal of Information Security*, pages 1–14, 2014. ISSN 1615-5262, 1615-5270. doi:10.1007/s10207-014-0242-0. [pages 14, 15, 20, 28, 128, 130, and 132]
- [112] L. Hao, C. G. Healey, and S. E. Hutchinson. Flexible Web Visualization for Alert-based Network Security Analytics. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, VizSec '13, pages 33–40, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517962. [pages 14, 15, 20, 28, and 70]
- [113] M. Hao, M. Marwah, S. Mittelstädt, H. Janetzko, D. Keim, U. Dayal, C. Bash, C. Felix, C. Patel, M. Hsu, Y. Chen, and M. Hund. Visual analytics of cyber physical data streams using spatio-temporal radial pixel visualization. In *Proc. SPIE*, volume 8654, pages 865404–865404–12, 2013. doi:10.1117/12.2002948. [pages 14, 20, 28, and 70]
- [114] L. Harrison and A. Lu. The future of security visualization: Lessons from network visualization. *IEEE Network*, 26(6):6–11, 2012. ISSN 0890-8044. doi:10.1109/MNET.2012.6375887. [pages 11 and 14]
- [115] L. Harrison, R. Spahn, M. Iannacone, E. Downing, and J. R. Goodall. NV: Nessus Vulnerability Visualization for the Web. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, VizSec '12, pages 25–32, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379694. [pages 14, 15, 20, 28, and 70]
- [116] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1303–1312, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi:10.1145/1518701.1518897. [page 59]

- [117] heise online. Bundestag-Hack war ein Phishing-Angriff über un.org (in German). URL <http://www.heise.de/newsticker/meldung/Bundestag-Hack-war-ein-Phishing-Angriff-ueber-un-org-2811847.html>. Accessed: 2015-12-01. [page 1]
- [118] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective Visualization of File System Access-Control. In J. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, volume 5210 of *Lecture Notes in Computer Science*, pages 18–25. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4. [page 137]
- [119] A. Herrero, E. Corchado, M. A. Pellicer, and A. Abraham. MOVIH-IDS: A Mobile-visualization Hybrid Intrusion Detection System. *Neurocomput.*, 72(13-15): 2775–2784, 2009. ISSN 0925-2312. doi:10.1016/j.neucom.2008.12.033. [pages 14, 15, 20, 28, and 105]
- [120] X. Hu and Z. M. Mao. Accurate Real-Time Identification of IP Prefix Hijacking. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 3–17, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2848-1. doi:10.1109/SP.2007.7. [pages 113, 115, and 124]
- [121] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. ELVIS: Extensible Log VISualization. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, VizSec '13, pages 9–16, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517959. [pages 14, 20, 28, 70, and 148]
- [122] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. CORGI: Combination, Organization and Reconstruction Through Graphical Interactions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, VizSec '14, pages 57–64, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671494. [pages 14, 20, 28, and 70]
- [123] IETF. IP Flow Information Export (IPFIX) Entities, . URL <http://www.iana.org/assignments/ipfix/ipfix.xhtml>. Accessed: 2015-10-01. [page 42]
- [124] IETF. RFC7011 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, . URL <https://tools.ietf.org/html/rfc7011>. Accessed: 2015-10-01. [page 41]
- [125] D. Inoue, M. Eto, K. Suzuki, M. Suzuki, and K. Nakao. DAEDALUS-VIZ: Novel Real-time 3d Visualization for Darknet Monitoring-based Alert System. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, VizSec '12, pages 72–79, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379700. [pages 14, 15, 20, 28, and 105]
- [126] J. Isacenkova, O. Thonnard, A. Costin, D. Balzarotti, and A. Francillon. Inside the SCAM Jungle: A Closer Look at 419 Scam Email Operations. In *Security and Privacy Workshops (SPW), 2013 IEEE*, pages 143–150, 2013. doi:10.1109/SPW.2013.15. [pages 137, 138, and 179]
- [127] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Moller. A Systematic Review on the Practice of Evaluating Visualization. *IEEE Transactions on*

- Visualization and Computer Graphics*, 19(12):2818–2827, 2013. ISSN 1077-2626. doi:[10.1109/TVCG.2013.126](https://doi.org/10.1109/TVCG.2013.126). [page 33]
- [128] ITU. ITU-T X.1205 Overview of cybersecurity. URL <https://www.itu.int/rec/T-REC-X.1205-200804-I>. Accessed: 2015-10-01. [page 2]
- [129] B. Jackson, D. Coffey, L. Thorson, D. Schroeder, A. M. Ellingson, D. J. Nuckley, and D. F. Keefe. Toward Mixed Method Evaluations of Scientific Visualizations and Design Process As an Evaluation Tool. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, pages 4:1–4:6, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1791-7. doi:[10.1145/2442576.2442580](https://doi.org/10.1145/2442576.2442580). [page 33]
- [130] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams. Cauldron mission-centric cyber situational awareness with defense in depth. In *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, pages 1339–1344, 2011. doi:[10.1109/MILCOM.2011.6127490](https://doi.org/10.1109/MILCOM.2011.6127490). [pages 14, 15, 20, 28, and 105]
- [131] J. Janies. Existence Plots: A Low-Resolution Time Series for Port Behavior Analysis. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, number 5210 in Lecture Notes in Computer Science, pages 161–168. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4, 978-3-540-85933-8. [pages 14, 15, 20, 28, and 60]
- [132] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics*, 16(6):927–34, 2010. ISSN 1077-2626. doi:[10.1109/TVCG.2010.162](https://doi.org/10.1109/TVCG.2010.162). [pages 59 and 64]
- [133] D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):141–150, 2016. ISSN 1077-2626. doi:[10.1109/TVCG.2015.2467553](https://doi.org/10.1109/TVCG.2015.2467553). [pages 5, 14, 20, 28, 106, 107, 108, 169, and 178]
- [134] K. Kancherla and S. Mukkamala. Image visualization based malware detection. In *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 40–44, 2013. doi:[10.1109/CICYBS.2013.6597204](https://doi.org/10.1109/CICYBS.2013.6597204). [pages 14, 15, 20, 28, 128, 130, and 132]
- [135] G. Kaur, V. Saxena, and J. Gupta. Anomaly Detection in network traffic and role of wavelets. In *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 7, pages V7–46–V7–51, 2010. doi:[10.1109/ICCET.2010.5485392](https://doi.org/10.1109/ICCET.2010.5485392). [page 61]
- [136] D. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002. ISSN 1077-2626. doi:[10.1109/2945.981847](https://doi.org/10.1109/2945.981847). [pages 27, 29, and 181]
- [137] S. Kent. Securing the Border Gateway Protocol: A Status Update. In *In Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, pages 2–3, 2003. [page 113]

- [138] E. Keogh, J. Lin, and A. Fu. HOT SAX: efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining*, pages 8 pp.–, 2005. doi:[10.1109/ICDM.2005.79](https://doi.org/10.1109/ICDM.2005.79). [page 60]
- [139] R. Kincaid and H. Lam. Line graph explorer: Scalable display of line graphs using Focus+Context. In *AVI*, pages 404–411, 2006. [page 60]
- [140] C. Kintzel, J. Fuchs, and F. Mansmann. Monitoring Large IP Spaces with ClockView. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11*, pages 2:1–2:10, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0679-9. doi:[10.1145/2016904.2016906](https://doi.org/10.1145/2016904.2016906). [pages 14, 15, 20, 28, 69, 70, 71, and 75]
- [141] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. ISBN 0-201-89684-2. [page 62]
- [142] H. Koike and K. Ohno. SnortView: Visualization System of Snort Logs. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04*, pages 143–147, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029232](https://doi.org/10.1145/1029208.1029232). [pages 14, 15, 20, 28, 104, 105, and 148]
- [143] H. Koike, K. Ohno, and K. Koizumi. Visualizing Cyber Attacks Using IP Matrix. In *Proceedings of the IEEE Workshops on Visualization for Computer Security, VIZSEC '05*, pages 11–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9477-1. doi:[10.1109/VIZSEC.2005.22](https://doi.org/10.1109/VIZSEC.2005.22). [pages 14, 15, 20, 28, and 105]
- [144] I. Kottenko and E. Novikova. Visualization of Security Metrics for Cyber Situation Awareness. In *2014 Ninth International Conference on Availability, Reliability and Security (ARES)*, pages 506–513, 2014. doi:[10.1109/ARES.2014.75](https://doi.org/10.1109/ARES.2014.75). [pages 14, 20, 28, and 70]
- [145] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, pages 42–49, 2005. doi:[10.1109/IAW.2005.1495932](https://doi.org/10.1109/IAW.2005.1495932). [pages 14, 15, 20, 28, and 105]
- [146] M. I. Krzywinski, J. E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009. doi:[10.1101/gr.092759.109](https://doi.org/10.1101/gr.092759.109). [page 138]
- [147] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: A prefix hijack alert system. In *Proc. USENIX Security Symposium*, 2006. [page 113]
- [148] M. Lad, D. Massey, and L. Zhang. Visualizing internet routing changes. *IEEE Transactions on Visualization and Computer Graphics*, pages 1450–1460, 2006. ISSN 1077-2626. [pages 14, 15, 20, 28, and 111]
- [149] K. Lakkaraju, W. Yurcik, and A. J. Lee. NVisionIP: Netflow Visualizations of System State for Security Situational Awareness. In *Proceedings of the*

- 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04, pages 65–72, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029219](https://doi.org/10.1145/1029208.1029219). [pages 14, 15, 20, 28, and 70]
- [150] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, 2012. ISSN 1077-2626. doi:[10.1109/TVCG.2011.279](https://doi.org/10.1109/TVCG.2011.279). [page 30]
- [151] J. Landstorfer, I. Herrmann, J.-E. Stange, M. Dork, and R. Wettach. Weaving a carpet from log entries: A network security visualization built with co-creation. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 73–82, 2014. doi:[10.1109/VAST.2014.7042483](https://doi.org/10.1109/VAST.2014.7042483). [pages 14, 20, 28, and 70]
- [152] M. Lanzemberger, S. Miksch, and M. Pohl. Exploring highly structured data: a comparative study of star diagrams and parallel coordinates. In *Ninth International Conference on Information Visualisation, 2005. Proceedings*, pages 312–320, 2005. doi:[10.1109/IV.2005.49](https://doi.org/10.1109/IV.2005.49). [page 27]
- [153] V. Lavigne and D. Gouin. Visual Analytics for cyber security and intelligence. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 11(2):175–199, 2014. ISSN 1548-5129, 1557-380X. doi:[10.1177/1548512912464532](https://doi.org/10.1177/1548512912464532). [page 14]
- [154] J. LeBlanc, M. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of the First IEEE Conference on Visualization, 1990. Visualization '90*, pages 230–237, 1990. doi:[10.1109/VISUAL.1990.146386](https://doi.org/10.1109/VISUAL.1990.146386). [page 27]
- [155] C. Lee, J. Trost, N. Gibbs, R. Beyah, and J. Copeland. Visual firewall: real-time network security monitor. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*, pages 129–136, 2005. doi:[10.1109/VIZSEC.2005.1532075](https://doi.org/10.1109/VIZSEC.2005.1532075). [pages 14, 15, 20, 28, and 105]
- [156] T. R. Leschke and C. Nicholas. Change-link 2.0: A Digital Forensic Tool for Visualizing Changes to Shadow Volume Data. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security, VizSec '13*, pages 17–24, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:[10.1145/2517957.2517960](https://doi.org/10.1145/2517957.2517960). [pages 14, 20, 28, and 70]
- [157] T. R. Leschke and A. T. Sherman. Change-Link: A Digital Forensic Tool for Visualizing Changes to Directory Trees. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 48–55, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:[10.1145/2379690.2379697](https://doi.org/10.1145/2379690.2379697). [pages 14, 20, 28, and 70]
- [158] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes. A survey of network flow applications. *Journal of Network and Computer Applications*, 36(2):567–581, 2013. ISSN 1084-8045. doi:[10.1016/j.jnca.2012.12.020](https://doi.org/10.1016/j.jnca.2012.12.020). [pages 10 and 14]
- [159] Q. Liao, A. Striegel, and N. Chawla. Visualizing Graph Dynamics and Similarity for Enterprise Network Security and Management. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security, VizSec*

- '10, pages 34–45, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0013-1. doi:[10.1145/1850795.1850799](https://doi.org/10.1145/1850795.1850799). [pages 14, 15, 20, 28, and 105]
- [160] Y. Livnat, J. Agutter, S. Moon, R. Erbacher, and S. Foresti. A visualization paradigm for network intrusion detection. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, pages 92–99, 2005. doi:[10.1109/IAW.2005.1495939](https://doi.org/10.1109/IAW.2005.1495939). [pages 14, 15, 20, 28, and 105]
- [161] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 95–102, Oct 2005. doi:[10.1109/INFVIS.2005.1532134](https://doi.org/10.1109/INFVIS.2005.1532134). [pages 14, 15, 20, 28, and 105]
- [162] A. Long, J. Saxe, and R. Gove. Detecting Malware Samples with Similar Image Sets. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 88–95, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:[10.1145/2671491.2671500](https://doi.org/10.1145/2671491.2671500). [pages 14, 15, 20, 28, 128, 130, and 131]
- [163] F. Mansman, L. Meier, and D. A. Keim. Visualization of Host Behavior for Network Security. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *VizSEC 2007, Mathematics and Visualization*, pages 187–202. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78242-1, 978-3-540-78243-8. [pages 14, 15, 20, 28, and 70]
- [164] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North. Visualizing large-scale IP traffic flows. In *Proceedings of 12th International Workshop Vision, Modeling, and Visualization (VMV 2007)*, 2007. [pages 14, 20, 28, 104, and 105]
- [165] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda. 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. doi:[10.1109/TVCG.2007.70522](https://doi.org/10.1109/TVCG.2007.70522). [pages 14, 15, 20, 28, 34, 71, 104, and 105]
- [166] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North. Visual Support for Analyzing Network Traffic and Intrusion Detection Events Using TreeMap and Graph Representations. In *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology, CHiMiT '09*, pages 3:19–3:28, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-572-7. doi:[10.1145/1641587.1641590](https://doi.org/10.1145/1641587.1641590). [pages 14, 15, 20, 28, 104, and 105]
- [167] F. Mansmann, F. Fischer, and D. A. Keim. Dynamic Visual Analytics – Facing the Real-Time Challenge. In J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, editors, *Expanding the Frontiers of Visual Analytics and Visualization*, pages 69–80. Springer London, 2012. ISBN 978-1-4471-2803-8 978-1-4471-2804-5. [pages 7, 145, and 146]
- [168] F. Mansmann, T. Göbel, and W. Cheswick. Visual Analysis of Complex Firewall Configurations. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 1–8, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:[10.1145/2379690.2379691](https://doi.org/10.1145/2379690.2379691). [pages 14, 15, 20, 28, 60, and 69]



- [169] F. Mansmann, M. Krstajic, F. Fischer, and E. Bertini. StreamSqueeze: A Dynamic Stream Visualization for Monitoring of Event Data. In *Proceedings of Conference on Visualization and Data Analysis (VDA '12)*, volume 8294, pages 829404–829404–12, 2012. doi:[10.1111/12.912372](https://doi.org/10.1111/12.912372). [pages 7, 14, 20, 28, 36, and 70]
- [170] R. Marty. *Applied Security Visualization*. Addison-Wesley Professional, 2008. ISBN 978-0-321-51010-5. [page 137]
- [171] J. E. McGrath. Methodology Matters: Doing Research in the Behavioral and Social Sciences. In R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, editors, *Human-computer Interaction*, pages 152–169. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995. ISBN 1-55860-246-1. [pages 30, 31, 32, and 33]
- [172] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: Interactive Visual Exploration of System Management Time-series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, pages 1483–1492, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi:[10.1145/1357054.1357286](https://doi.org/10.1145/1357054.1357286). [pages 14, 15, 20, 28, 70, and 148]
- [173] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. PortVis: A Tool for Port-based Detection of Security Events. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04*, pages 73–81, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029220](https://doi.org/10.1145/1029208.1029220). [pages 14, 15, 20, 28, 60, and 107]
- [174] K.-R. Müller. *IT-Sicherheit mit System (in German)*. Springer Fachmedien Wiesbaden, Wiesbaden, 2014. ISBN 978-3-658-04333-9 978-3-658-04334-6. [page 2]
- [175] A. Moser, C. Kruegel, and E. Kirda. Exploring Multiple Execution Paths for Malware Analysis. In *IEEE Symposium on Security and Privacy, 2007. SP '07*, pages 231–245, 2007. doi:[10.1109/SP.2007.17](https://doi.org/10.1109/SP.2007.17). [pages 22 and 127]
- [176] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. Malware Images: Visualization and Automatic Classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11*, pages 4:1–4:7, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0679-9. doi:[10.1145/2016904.2016908](https://doi.org/10.1145/2016904.2016908). [pages 14, 15, 20, 28, 128, 130, and 132]
- [177] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang. A Comparative Assessment of Malware Classification Using Binary Texture Analysis and Dynamic Analysis. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, AISec '11*, pages 21–30, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1003-1. doi:[10.1145/2046684.2046689](https://doi.org/10.1145/2046684.2046689). [page 132]
- [178] J. Nielsen. Usability Inspection Methods. In *Conference Companion on Human Factors in Computing Systems, CHI '94*, pages 413–414, New York, NY, USA, 1994. ACM. ISBN 0-89791-651-4. doi:[10.1145/259963.260531](https://doi.org/10.1145/259963.260531). [page 33]
- [179] T. Nunnally, P. Chi, K. Abdullah, A. Uluagac, J. Copeland, and R. Beyah. P3d: A parallel 3d coordinate visualization for advanced network scans. In *2013 IEEE International Conference on Communications (ICC)*, pages 2052–2057, 2013. doi:[10.1109/ICC.2013.6654828](https://doi.org/10.1109/ICC.2013.6654828). [pages 14, 15, 20, 28, and 105]

- [180] K. Nyarko, T. Capers, C. Scott, and K. Ladeji-Osias. Network intrusion visualization with NIVA, an intrusion detection visual analyzer with haptic integration. In *10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings*, pages 277–284, 2002. doi:10.1109/HAPTIC.2002.998969. [pages 14, 15, 20, 28, and 105]
- [181] J. Oberheide, M. Karir, and D. Blazakis. VAST: Visualizing Autonomous System Topology. In *Proceedings of the 3rd International Workshop on Visualization for Computer Security, VizSEC '06*, pages 71–80, New York, NY, USA, 2006. ACM. ISBN 1-59593-549-5. doi:10.1145/1179576.1179592. [pages 14, 15, 20, 28, 111, and 112]
- [182] I.-V. Onut and A. A. Ghorbani. SVision: A novel visual network-anomaly identification technique. *Computers & Security*, 26(3):201–212, 2007. ISSN 0167-4048. doi:10.1016/j.cose.2006.10.001. [pages 14, 15, 20, 28, and 105]
- [183] J. Ortiz-Ubarri, H. Ortiz-Zuazaga, A. Maldonado, E. Santos, and J. Grullon. Toa: A Web Based Network Flow Data Monitoring System at Scale. In *2015 IEEE International Congress on Big Data (BigData Congress)*, pages 438–443, 2015. doi:10.1109/BigDataCongress.2015.71. [pages 14, 20, 28, and 105]
- [184] OSSEC. Open Source Host-based Intrusion Detection System. URL <http://www.ossec.net/>. Accessed: 2015-10-01. [page 158]
- [185] T. Panas. Signature Visualization of Software Binaries. In *Proceedings of the 4th ACM Symposium on Software Visualization, SoftVis '08*, pages 185–188, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-112-5. doi:10.1145/1409720.1409749. [pages 14, 15, 20, 28, 128, 130, 132, and 133]
- [186] S. Papadopoulos, K. Moustakas, and D. Tzovaras. Hierarchical Visualization of BGP Routing Changes Using Entropy Measures. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, editors, *Advances in Visual Computing*, number 7432 in Lecture Notes in Computer Science, pages 696–705. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33190-9 978-3-642-33191-6. [pages 14, 15, 20, 28, and 111]
- [187] S. Papadopoulos, G. Theodoridis, and D. Tzovaras. BGPfuse: Using Visual Feature Fusion for the Detection and Attribution of BGP Anomalies. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security, VizSec '13*, pages 57–64, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517965. [pages 14, 20, 28, and 111]
- [188] A. Paturi, M. Cherukuri, J. Donahue, and S. Mukkamala. Mobile malware visual analytics and similarities of Attack Toolkits (Malware gene analysis). In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 149–154, 2013. doi:10.1109/CTS.2013.6567221. [pages 14, 15, 20, 28, 128, 130, and 133]
- [189] J. Pearlman and P. Rheingans. Visualizing Network Security Events Using Compound Glyphs from a Service-Oriented Perspective. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *VizSEC 2007, Mathematics and Visualization*, pages 131–146. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78242-1, 978-3-540-78243-8. [pages 14, 15, 20, 28, and 70]

- [190] D. B. Percival and A. T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000. ISBN 978-0-521-64068-8. [page 61]
- [191] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd. Visual analysis of network flow data with timelines and event plots. *VizSEC 2007*, 2008. [pages 14, 15, 20, 28, and 70]
- [192] R. Pickett and G. Grinstein. Iconographic Displays For Visualizing Multidimensional Data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, 1988*, volume 1, pages 514–519, 1988. doi:10.1109/ICSMC.1988.754351. [page 27]
- [193] W. A. Pike, C. Scherrer, and S. Zabriskie. Putting Security in Context: Visual Correlation of Network Activity with Real-World Information. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *VizSEC 2007*, Mathematics and Visualization, pages 203–220. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78242-1 978-3-540-78243-8. [pages 14, 15, 20, 28, and 41]
- [194] J. Qiu and L. Gao. Detecting Bogus BGP Route Information: Going Beyond Prefix Hijacking. Technical report, In Proc. SecureComm, 2007. [page 113]
- [195] D. Quist and L. Liebrock. Visualizing compiled executables for malware analysis. In *6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009*, pages 27–32, 2009. doi:10.1109/VIZSEC.2009.5375539. [pages 14, 15, 20, 28, 128, and 130]
- [196] D. A. Quist and L. M. Liebrock. Reversing Compiled Executables for Malware Analysis via Visualization. *Information Visualization*, 10(2):117–126, 2011. ISSN 1473-8716, 1473-8724. doi:10.1057/ivs.2010.11. [pages 14, 15, 20, 28, 128, and 130]
- [197] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. *ACM SIGCOMM Computer Communication Review*, 36(4):291, 2006. ISSN 01464833. doi:10.1145/1151659.1159947. [page 124]
- [198] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson. IDGraphs: intrusion detection and analysis using histograms. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*, pages 39–46, 2005. doi:10.1109/VIZSEC.2005.1532064. [pages 14, 15, 20, 28, 104, and 105]
- [199] P. Ren, J. Kristoff, and B. Gooch. Visualizing DNS Traffic. In *Proceedings of the 3rd International Workshop on Visualization for Computer Security, VizSEC '06*, pages 23–30, New York, NY, USA, 2006. ACM. ISBN 1-59593-549-5. doi:10.1145/1179576.1179582. [pages 14, 15, 20, 28, and 105]
- [200] J. C. Roberts, D. A. Keim, T. Hanratty, R. R. Rowlingson, R. Walker, M. Hall, Z. Jacobson, V. Lavigne, C. Rooney, and M. Varga. From Ill-Defined Problems to Informed Decisions. *EuroVis Workshop on Visual Analytics (2014)*, 2014. [page 146]
- [201] C. Rohrdantz, D. Oelke, M. Krstajic, and F. Fischer. Real-Time Visualization of Streaming Text Data: Tasks and Challenges. In *Workshop on Interactive Visual Text Analytics for Decision-Making at the IEEE VisWeek 2011*, 2011. [page 7]

- [202] Routeviews. University of Oregon Route Views Project. URL <http://www.routeviews.org/>. Accessed: 2015-10-01. [page 115]
- [203] F. Roveta, G. Caviglia, L. Di Mario, S. Zanero, F. Maggi, and P. Ciuccarelli. BURN: Baring Unknown Rogue Networks. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11*, pages 6:1–6:10, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0679-9. doi:10.1145/2016904.2016910. [pages 14, 20, 28, 36, and 136]
- [204] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *INFOVIS*, page 23, 2005. [page 59]
- [205] E. B.-N. Sanders and P. J. Stappers. Co-creation and the new landscapes of design. *CoDesign*, 4(1):5–18, 2008. ISSN 1571-0882. doi:10.1080/15710880701875068. [page 116]
- [206] J. Saxe, D. Mentis, and C. Greamo. Visualization of Shared System Call Sequence Relationships in Large Malware Corpora. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 33–40, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379695. [pages 14, 15, 20, 28, 128, 130, 131, and 134]
- [207] B. Schneider, C. Acevedo, J. Buchmüller, F. Fischer, and D. A. Keim. Visual analytics for inspecting the evolution of a graph over time: Pattern discovery in a communication network. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 169–170, 2015. doi:10.1109/VAST.2015.7347664. [page 8]
- [208] T. Schreck, D. A. Keim, and F. Mansmann. Regular TreeMap Layouts for Visual Analysis of Hierarchical Data. In *Proceedings of the Spring Conference on Computer Graphics (SCCG'2006)*, Casta Papiernicka, Slovak Republic, 2006. ACM Siggraph. [pages 71 and 77]
- [209] C. Scott, K. Nyarko, T. Capers, and J. Ladeji-Osias. Network Intrusion Visualization with NIVA, an Intrusion Detection Visual and Haptic Analyzer. *Information Visualization*, 2(2):82–94, 2003. ISSN 1473-8716. doi:10.1057/palgrave.ivs.9500044. [pages 14, 15, 20, 28, and 105]
- [210] SEMVAST. Visual Analytics Benchmark Repository - VAST Challenge 2013-2014 Dataset. URL <http://hcil2.cs.umd.edu/newvarepository/>. Accessed: 2015-10-01. [pages 49, 50, 52, 54, 55, and 95]
- [211] I. Shafer, K. Ren, V. N. Boddeti, Y. Abe, G. R. Ganger, and C. Faloutsos. RainMon: An Integrated Approach to Mining Bursty Timeseries Monitoring Data. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1158–1166, 2012. doi:10.1145/2339530.2339711. [pages 14, 20, 28, 60, and 70]
- [212] Shaid and Maarof. Malware behavior image for malware variant identification. In *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, pages 238–243, 2014. doi:10.1109/ISBAST.2014.7013128. [pages 14, 15, 20, 28, 128, 130, 132, and 133]

- [213] Shaid and Maarof. Malware behaviour visualization. *Jurnal Teknologi*, 70(5): 25–33, 2014. doi:[10.11113/jt.v70.3512](https://doi.org/10.11113/jt.v70.3512). [pages 14, 15, 20, 28, 128, 130, 132, and 133]
- [214] J. Shearer, K.-L. Ma, and T. Kohlenberg. BGPeep: An IP-Space Centered View for Internet Routing Data. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *Visualization for Computer Security*, number 5210 in Lecture Notes in Computer Science, pages 95–110. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85931-4 978-3-540-85933-8. [pages 15, 20, 28, 111, and 113]
- [215] H. Shiravi, A. Shiravi, and A. A. Ghorbani. IDS Alert Visualization and Monitoring through Heuristic Host Selection. *Security*, pages 445–458, 2010. [pages 14, 15, 20, 28, and 105]
- [216] H. Shiravi, A. Shiravi, and A. Ghorbani. A Survey of Visualization Systems for Network Security. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1313–1329, 2012. ISSN 1077-2626. doi:[10.1109/TVCG.2011.144](https://doi.org/10.1109/TVCG.2011.144). [pages 10, 11, 12, 14, 15, 18, 19, 20, 21, 22, 23, 104, 139, 145, 165, and 181]
- [217] B. Shneiderman. Tree Visualization with Tree-Maps: A 2-D Space-Filling Approach. *ACM Transactions on Graphics*, 11:92–99, 1991. [pages 27 and 71]
- [218] 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. [page 117]
- [219] B. Shneiderman and C. Plaisant. Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM. ISBN 1-59593-562-2. doi:[10.1145/1168149.1168158](https://doi.org/10.1145/1168149.1168158). [pages 31 and 139]
- [220] S. Simon, S. Mittelstädt, D. A. Keim, and M. Sedlmair. Bridging the Gap of Domain and Visualization Experts with a Liaison. In E. Bertini, J. Kennedy, and E. Puppo, editors, *Eurographics Conference on Visualization (EuroVis) - Short Papers*. The Eurographics Association, 2015. doi:[10.2312/eurovisshort.20151137](https://doi.org/10.2312/eurovisshort.20151137). [page 116]
- [221] H. Song, C. Muelder, and K.-L. Ma. Crucial Nodes Centric Visual Monitoring and Analysis of Computer Networks. In *2012 International Conference on Cyber Security (CyberSecurity)*, pages 16–23, 2012. doi:[10.1109/CyberSecurity.2012.9](https://doi.org/10.1109/CyberSecurity.2012.9). [pages 14, 20, 28, and 70]
- [222] D. Spirin. Prefix hijacking by Michael Lindsay via Internap. URL <http://mailman.nanog.org/pipermail/nanog/2011-August/039379.html>. Accessed: 2011-08-30. [page 125]
- [223] D. Staheli, T. Yu, R. J. Crouser, S. Damodaran, K. Nam, D. O’Gwynn, S. McKenna, and L. Harrison. Visualization Evaluation for Cyber Security: Trends and Future Directions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec ’14*, pages 49–56, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:[10.1145/2671491.2671492](https://doi.org/10.1145/2671491.2671492). [pages 11, 30, 31, 32, 34, and 36]

- [224] J.-E. Stange, M. Dörk, J. Landstorfer, and R. Wettach. Visual Filter: Graphical Exploration of Network Security Log Files. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, VizSec '14, pages 41–48, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671503. [pages 14, 20, 28, and 70]
- [225] F. Stoffel and F. Fischer. Using a Knowledge Graph Data Structure to Analyze Text Documents (VAST Challenge 2014 MC1). In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 331–332, 2014. doi:10.1109/VAST.2014.7042551. [page 8]
- [226] F. Stoffel, F. Fischer, and D. A. Keim. Finding Anomalies in Time-Series using Visual Correlation for Interactive Root Cause Analysis. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, VizSec '13, pages 65–72, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2173-0. doi:10.1145/2517957.2517966. [pages 5, 14, 20, 28, 58, 62, 64, 66, and 175]
- [227] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda. FIRE: FInding Rogue nEtworks. In *Computer Security Applications Conference, 2009. ACSAC '09. Annual*, pages 231–240, 2009. doi:10.1109/ACSAC.2009.29. [page 136]
- [228] D. Streeb, U. Schlegel, J. Buchmüller, F. Fischer, and D. A. Keim. Using visual analytics to analyze movement and action patterns. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 171–172, 2015. doi:10.1109/VAST.2015.7347665. [page 8]
- [229] Symantec. The Elderwood Project, . URL <http://www.symantec.com/connect/blogs/elderwood-project>. Accessed: 2015-10-01. [page 140]
- [230] Symantec. Symantec Internet Security Threat Report 2012, . URL <http://www.symantec.com/threatreport/>. Accessed: 2015-02-24. [pages 125 and 126]
- [231] Symantec. Symantec Internet Security Threat Report 2015, . URL <http://www.symantec.com/threatreport/>. Accessed: 2015-10-01. [pages 1 and 127]
- [232] Symantec. Symantec.cloud, . URL <http://www.symanteccloud.com/>. Accessed: 2015-02-24. [page 115]
- [233] C. Systems. RFC3954 - Cisco Systems NetFlow Services Export Version 9, . URL <https://tools.ietf.org/html/rfc3954>. Accessed: 2015-10-01. [page 41]
- [234] C. Systems. RFC5101 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, . URL <https://tools.ietf.org/html/rfc5101>. Accessed: 2015-10-01. [page 41]
- [235] G. Tadda and J. Salerno. Overview of Cyber Situation Awareness. In *Cyber Situational Awareness*, volume 46 of *Advances in Information Security*, pages 15–35. Springer US, 2010. ISBN 978-1-4419-0139-2. [pages 10 and 148]
- [236] M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima. A Method to Detect Prefix Hijacking by Using Ping Tests. In *APNOMS '08: Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management*, pages 390–398, Beijing, China, 2008. Springer-Verlag. ISBN 978-3-540-88622-8. [pages 113 and 116]

- [237] T. Takada and H. Koike. Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs. In *Sixth International Conference on Information Visualisation, 2002. Proceedings*, pages 570–576, 2002. doi:[10.1109/IV.2002.1028831](https://doi.org/10.1109/IV.2002.1028831). [pages 14, 15, 20, 28, and 70]
- [238] R. Tamassia, B. Palazzi, and C. Papamanthou. Graph Drawing for Security Visualization. In I. G. Tollis and M. Patrignani, editors, *Graph Drawing*, number 5417 in Lecture Notes in Computer Science, pages 2–13. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00218-2 978-3-642-00219-9. [page 10]
- [239] T. Taylor, S. Brooks, and J. McHugh. NetBytes Viewer: An Entity-Based NetFlow Visualization Utility for Identifying Intrusive Behavior. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *VizSEC 2007, Mathematics and Visualization*, pages 101–114. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78242-1, 978-3-540-78243-8. [pages 14, 15, 20, 28, and 60]
- [240] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh. FloVis: Flow Visualization System. In *Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications Technology*, pages 186–198, 2009. doi:[10.1109/CATCH.2009.18](https://doi.org/10.1109/CATCH.2009.18). [pages 14, 15, 20, 28, and 41]
- [241] S. T. Teoh, K. L. Ma, S. F. Wu, A. Mankin, D. Massey, and X. Zhao. *ELISHA: A Visual-Based Anomaly Detection System for the BGP Routing Protocol*. 2002. [pages 14, 15, 20, 28, 111, and 112]
- [242] S. T. Teoh, K.-l. Ma, S. F. Wu, and K. Words. A Visual Technique for Internet Anomaly Detection. In *IASTED International Conference on Computer Graphics and Imaging (CGIM '02), IASTED*. ACTA Press, 2002. [pages 14, 20, 28, and 111]
- [243] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao. Case study: interactive visualization for internet security. In *Proceedings of the conference on Visualization '02, VIS '02*, pages 505–508, Boston, Massachusetts, 2002. IEEE Computer Society. ISBN 0-7803-7498-3. [pages 14, 15, 20, 28, and 111]
- [244] S. T. Teoh, K.-L. Ma, S. Wu, and T. Jankun-Kelly. Detecting flaws and intruders with visual data analysis. *IEEE Computer Graphics and Applications*, 24(5): 27–35, 2004. ISSN 0272-1716. doi:[10.1109/MCG.2004.26](https://doi.org/10.1109/MCG.2004.26). [pages 14, 15, 20, 28, 105, and 111]
- [245] S. T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma, and S. F. Wu. Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 35–44, Washington DC, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029215](https://doi.org/10.1145/1029208.1029215). [pages 14, 15, 20, 28, 111, and 112]
- [246] S. T. Teoh, S. Ranjan, A. Nucci, and C.-N. Chuah. BGP eye: a new visualization tool for real-time detection and analysis of BGP anomalies. In *VizSEC '06: Proceedings of the 3rd international workshop on Visualization for computer security*, pages 81–90, Alexandria, Virginia, USA, 2006. ACM. ISBN 1-59593-549-5. doi:[10.1145/1179576.1179593](https://doi.org/10.1145/1179576.1179593). [pages 14, 15, 20, 28, 111, and 113]

- [247] J. Thomas, K. Cook, I. Electrical, and E. Engineers. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society, 2005. ISBN 0769523234. [page 2]
- [248] O. Thonnard and M. Dacier. A Strategic Analysis of Spam Botnets Operations. In *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, CEAS '11*, pages 162–171, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0788-8. doi:10.1145/2030376.2030395. [page 137]
- [249] O. Thonnard, W. Mees, and M. Dacier. On a Multicriteria Clustering Approach for Attack Attribution. *SIGKDD Explor. Newsl.*, 12(1):11–20, 2010. ISSN 1931-0145. doi:10.1145/1882471.1882474. [pages 22, 38, 137, and 168]
- [250] O. Thonnard, L. Bilge, G. O’Gorman, S. Kiernan, and M. Lee. Industrial Espionage and Targeted Attacks: Understanding the Characteristics of an Escalating Threat. In D. Balzarotti, S. Stolfo, and M. Cova, editors, *Research in Attacks, Intrusions, and Defenses*, volume 7462 of *Lecture Notes in Computer Science*, pages 64–85. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33337-8. [page 137]
- [251] Tran Khanh Dang and Tran Tri Dang. A survey on security visualization techniques for web information systemsnull. *International Journal of Web Information Systems*, 9(1):6–31, 2013. ISSN 1744-0084. doi:10.1108/17440081311316361. [page 11]
- [252] P. Trinius, T. Holz, J. Gobel, and F. Freiling. Visual analysis of malware behavior using treemaps and thread graphs. In *6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009*, pages 33–38, 2009. doi:10.1109/VIZSEC.2009.5375540. [pages 14, 15, 20, 28, 128, 130, and 137]
- [253] O. Tsigkas, O. Thonnard, and D. Tzovaras. Visual Spam Campaigns Analysis Using Abstract Graphs Representation. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 64–71, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:10.1145/2379690.2379699. [pages 14, 20, 28, and 136]
- [254] Vaadin. Java Web Application Framework. URL <http://www.vaadin.com/>. Accessed: 2009-10-01. [page 83]
- [255] J. van Wijk. Evaluation: A Challenge for Visual Analytics. *Computer*, 46(7): 56–60, 2013. ISSN 0018-9162. doi:10.1109/MC.2013.151. [pages 33, 36, 138, and 157]
- [256] J. Van Wijk and H. Van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 73–78. IEEE, 1999. [page 71]
- [257] P.-A. Vervier and O. Thonnard. SpamTracer: How stealthy are spammers? In *2013 Proceedings IEEE INFOCOM*, pages 3477–3482, 2013. doi:10.1109/INFCOM.2013.6567184. [page 114]
- [258] VIS-SENSE. Visual Analytic Representation of Large Datasets for Enhancing Network Security. URL <http://www.vis-sense.eu/>. Accessed: 2015-10-01. [page 48]



- [259] VizSec. IEEE Symposium on Visualization for Cyber Security (VizSec). URL <http://vizsec.org/>. Accessed: 2015-10-03. [page 12]
- [260] C. Wagner, G. Wagener, R. State, A. Dulaunoy, and T. Engel. PeekKernelFlows: Peeking into IP Flows. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security, VizSec '10*, pages 52–57, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0013-1. doi:10.1145/1850795.1850801. [pages 14, 20, 28, and 105]
- [261] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner. A Survey of Visualization Systems for Malware Analysis. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*, Italy (Cagliari), 2015. The Eurographics Association. doi:10.2312/eurovisstar.20151114. [pages 5, 12, 13, 14, 15, 29, 127, and 167]
- [262] F. Waibel. Das Internet-Analyse-System (IAS) als Komponente einer IT-Sicherheitsarchitektur (in German). In *Sichere Wege in der vernetzten Welt - Tagungsband zum 11. Deutschen IT-Sicherheitskongress (in German)*, pages 281–296. SecuMedia Verlag, 2009. ISBN 978-3-922746-97-3. [page 59]
- [263] S. Walton, E. Maguire, and M. Chen. Multiple Queries with Conditional Attributes (QCATs) for Anomaly Detection and Visualization. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 17–24, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671502. [pages 14, 20, 28, and 70]
- [264] W. Wang, B. Yang, and V. Chen. A visual analytics approach to detecting server redirections and data exfiltration. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 13–18, 2015. doi:10.1109/ISI.2015.7165932. [pages 14, 20, 28, and 70]
- [265] F. Wanner, A. Stoffel, D. Jäckle, B. C. Kwon, A. Weiler, and D. A. Keim. State-of-the-Art Report of Visual Analysis for Event Detection in Text Data Streams. In *EuroVis - STARs*, pages 125–139, Swansea, UK, 2014. Eurographics Association. [page 148]
- [266] M. O. Ward. Multivariate Data Glyphs: Principles and Practice. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pages 179–198. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-33036-3 978-3-540-33037-0. [page 71]
- [267] T. Wüchner, A. Pretschner, and M. Ochoa. DAVAST: Data-centric System Level Activity Visualization. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security, VizSec '14*, pages 25–32, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2826-5. doi:10.1145/2671491.2671499. [pages 15, 20, 28, 128, and 130]
- [268] K. Wetzel. Pebbles - Using Circular Treemaps to Visualize Disk Usage. URL <http://lip.sourceforge.net/ctreemap.html>. Accessed: 2015-10-01. [page 71]
- [269] M. Whiting, K. Cook, C. L. Paul, K. Whitley, G. Grinstein, B. Nebesh, K. Liggett, M. Cooper, and J. Fallon. VAST Challenge 2013: Situation Awareness and Prospective Analysis. In *IEEE VAST 2013*, 2013. [pages 6, 43, 48, 52, and 57]

- [270] M. Whiting, K. Cook, G. Grinstein, K. Liggett, M. Cooper, J. Fallon, and M. Morin. VAST Challenge 2014: The Kronos Incident. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 295–300, 2014. doi:[10.1109/VAST.2014.7042536](https://doi.org/10.1109/VAST.2014.7042536). [pages 6, 162, and 163]
- [271] T. Wong, V. Jacobson, and C. Alaettinoglu. Internet routing anomaly detection and visualization. In *International Conference on Dependable Systems and Networks, 2005. DSN 2005. Proceedings*, pages 172–181, 2005. doi:[10.1109/DSN.2005.57](https://doi.org/10.1109/DSN.2005.57). [pages 14, 15, 20, 28, and 111]
- [272] Y. Wu and R. H. C. Yap. Experiments with Malware Visualization. In U. Flegel, E. Markatos, and W. Robertson, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, number 7591 in Lecture Notes in Computer Science, pages 123–133. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-37299-5, 978-3-642-37300-8. [pages 14, 15, 20, 28, 128, 130, and 132]
- [273] L. Xiao, J. Gerth, and P. Hanrahan. Enhancing Visual Analysis of Network Traffic Using a Knowledge Representation. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 107–114, 2006. doi:[10.1109/VAST.2006.261436](https://doi.org/10.1109/VAST.2006.261436). [pages 14, 15, 20, 28, 104, and 105]
- [274] Z. Xie, M. O. Ward, and E. A. Rundensteiner. Visual Exploration of Stream Pattern Changes Using a Data-Driven Framework. In *Advances in Visual Computing*, volume 6454 of *Lecture Notes in Computer Science*, pages 522–532. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17273-1. [page 148]
- [275] C. L. Yee, L. L. Chuan, M. Ismail, and N. Zainal. A static and dynamic visual debugger for malware analysis. In *2012 18th Asia-Pacific Conference on Communications (APCC)*, pages 765–769, 2012. doi:[10.1109/APCC.2012.6388211](https://doi.org/10.1109/APCC.2012.6388211). [pages 14, 15, 20, 28, 128, and 130]
- [276] A. Yelizarov and D. Gamayunov. Visualization of complex attacks and state of attacked network. In *6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009*, pages 1–9, 2009. doi:[10.1109/VIZSEC.2009.5375527](https://doi.org/10.1109/VIZSEC.2009.5375527). [pages 14, 15, 20, 28, and 105]
- [277] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. VisFlowConnect: Netflow Visualizations of Link Relationships for Security Situational Awareness. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04*, pages 26–34, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029214](https://doi.org/10.1145/1029208.1029214). [pages 14, 15, 20, 28, and 41]
- [278] I. Yoo. Visualizing Windows Executable Viruses Using Self-organizing Maps. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC '04*, pages 82–89, New York, NY, USA, 2004. ACM. ISBN 1-58113-974-8. doi:[10.1145/1029208.1029222](https://doi.org/10.1145/1029208.1029222). [pages 14, 15, 20, 28, 128, 130, and 133]
- [279] T. Yu, R. Lippmann, J. Riordan, and S. Boyer. EMBER: A Global Perspective on Extreme Malicious Behavior. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security, VizSec '10*, pages 1–12, New York,

- NY, USA, 2010. ACM. ISBN 978-1-4503-0013-1. doi:[10.1145/1850795.1850796](https://doi.org/10.1145/1850795.1850796). [pages [14](#), [20](#), [28](#), [135](#), and [136](#)]
- [280] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica. Discretized Streams: An Efficient and Fault-tolerant Model for Stream Processing on Large Clusters. In *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'12, pages 10–10, Berkeley, CA, USA, 2012. USENIX Association. [pages [146](#) and [163](#)]
- [281] H. Zhang, M. Sun, D. D. Yao, and C. North. Visualizing Traffic Causality for Analyzing Network Anomalies. In *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, IWSPA '15, pages 37–42, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3341-2. doi:[10.1145/2713579.2713583](https://doi.org/10.1145/2713579.2713583). [pages [14](#), [20](#), [28](#), and [70](#)]
- [282] Y. Zhang, Y. Xiao, M. Chen, J. Zhang, and H. Deng. A survey of security visualization for computer network logs. *Security and Communication Networks*, 5(4):404–421, 2012. ISSN 1939-0122. doi:[10.1002/sec.324](https://doi.org/10.1002/sec.324). [pages [10](#) and [14](#)]
- [283] Z. Zhang, Y. Zhang, Y. Charlie, H. Z. Morley, and M. R. Bush. iSPY: Detecting IP Prefix Hijacking on My Own. In *In Proc. ACM SIGCOMM*, 2008. [page [113](#)]
- [284] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory Analysis of Time-Series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011. ISSN 1077-2626. doi:[10.1109/TVCG.2011.195](https://doi.org/10.1109/TVCG.2011.195). [page [59](#)]
- [285] Y. Zhao, F. Zhou, and X. Fan. A Real-time Visualization Framework for IDS Alerts. In *Proceedings of the 5th International Symposium on Visual Information Communication and Interaction*, VINCI '12, pages 11–17, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1782-5. doi:[10.1145/2397696.2397698](https://doi.org/10.1145/2397696.2397698). [pages [14](#), [15](#), [20](#), [28](#), and [105](#)]
- [286] Y. Zhao, F. Zhou, X. Fan, X. Liang, and Y. Liu. IDSRadar: a real-time visualization framework for IDS alerts. *Science China Information Sciences*, 56(8):1–12, 2013. ISSN 1674-733X, 1869-1919. doi:[10.1007/s11432-013-4891-9](https://doi.org/10.1007/s11432-013-4891-9). [pages [14](#), [15](#), [20](#), [28](#), and [105](#)]
- [287] Y. Zhao, X. Liang, X. Fan, Y. Wang, M. Yang, and F. Zhou. MVSec: multi-perspective and deductive visual analytics on heterogeneous network security data. *Journal of Visualization*, 17(3):181–196, 2014. ISSN 1343-8875, 1875-8975. doi:[10.1007/s12650-014-0213-6](https://doi.org/10.1007/s12650-014-0213-6). [pages [14](#), [20](#), [28](#), and [105](#)]
- [288] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. *ACM SIGCOMM Computer Communication Review*, 37(4):277, 2007. ISSN 01464833. doi:[10.1145/1282427.1282412](https://doi.org/10.1145/1282427.1282412). [pages [113](#) and [116](#)]
- [289] F. Zhou, R. Shi, Y. Zhao, Y. Huang, and X. Liang. NetSecRadar: A Visualization System for Network Security Situational Awareness. In G. Wang, I. Ray, D. Feng, and M. Rajarajan, editors, *Cyberspace Safety and Security*, number 8300 in Lecture Notes in Computer Science, pages 403–416. Springer International Publishing, 2013. ISBN 978-3-319-03583-3 978-3-319-03584-0. [pages [14](#), [20](#), [28](#), and [105](#)]

- [290] W. Zhuo and Y. Nadjin. MalwareVis: Entity-based Visualization of Malware Network Traces. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security, VizSec '12*, pages 41–47, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1413-8. doi:[10.1145/2379690.2379696](https://doi.org/10.1145/2379690.2379696). [pages [14](#), [15](#), [20](#), [28](#), [128](#), and [130](#)]