

Bram Cappers

Interactive Visualization of Event logs for Cybersecurity

Bram Cappers

Interactive Visualization of Event logs for Cybersecurity

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de rector magnificus prof. dr. ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College
voor Promoties, in het openbaar te verdedigen op
dinsdag 4 december 2018 om 16:00 uur

door

Bram Cornelis Maria Cappers

geboren te Heerlen.

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

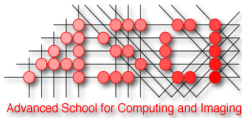
voorzitter:	prof. dr. J.J. Lukkien
1 ^e promotor:	prof. dr. ir. J.J. van Wijk
2 ^e promotor:	prof. dr. S. Etalle
leden:	prof. dr. M. Pechenizkiy
	prof. dr.-ing. J. Kohlhammer (Technische Universität Darmstadt)
	dr. M.A. Westenberg
	dr. L.T. Harrison (Worcester Polytechnic Institute Massachusetts)
	prof. dr. A.C. Telea (Rijksuniversiteit Groningen)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Colophon



The work in this dissertation is part of the NWO Cyber Security program with project number 628.001.004 (SpySpot: Visualization and Deep Protocol Analysis to Detect Cyber Espionage and Targeted Malware), which is financed by the Netherlands Organisation for Scientific Research (NWO)



This work was carried out in the ASCI graduate school. ASCI dissertation series number 398

Printed by: ProefschriftMaken

Cover design: Bram C.M. Cappers

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN 978-94-6380-043-3



An electronic version of this dissertation is available at

<http://repository.tue.nl> and
<http://www.bramcappers.nl>

Copyright © 2018 by B.C.M Cappers. All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Errors, like straws, upon the surface flow; He who would search for pearls, must dive below.

John Dryden

Contents

Colophon	iii
Preface	ix
1 Introduction	3
1.1 Motivation	3
1.2 Computer Networks	3
1.3 Multivariate Event Data.	4
1.4 The role of Visual Analytics.	5
1.5 Objective.	7
1.6 Outline & Contributions	7
1.7 Publications	10
2 Background	11
2.1 Data science	12
2.2 Data visualization	12
2.3 Users, Tasks, and Problems	15
2.4 Anomaly detection	16
2.5 Network Intrusion Detection	17
2.6 Advanced Persistent Threats	19
2.7 Network traffic	20
2.7.1 Non-intrusive vs. Intrusive Network Recording	20
2.7.2 Deep Packet Inspection.	21
2.8 Security visualization	25
2.9 Taxonomy security visualization	26
2.9.1 Host vs. Network events	26
2.9.2 Alert vs. Non-alert	27
2.9.3 Internal vs. External	28
2.9.4 Taxonomy Overview	29
2.9.5 Taxonomy Event visualization	29
2.9.6 Sequences vs. Records	31
2.9.7 Univariate vs. Multivariate	32
2.9.8 Low vs. High-dimensional	33
2.9.9 Taxonomy Overview	34
2.10 Bridging gaps between Event and Security visualization	35
3 Monitoring Multivariate Event Collections	37
3.1 Monitoring Multivariate Event Collections	38
3.2 Semantic Network traffic Analysis through Projection and Selection.	38

3.3	Related Work.	39
3.3.1	Data	39
3.3.2	Visualization	40
3.4	Problem statement	41
3.4.1	Data acquisition	42
3.5	SNAPS: Selection and Projection	43
3.5.1	Pixel viewer	44
3.5.2	Time view	47
3.5.3	Attribute view	48
3.6	Classification	49
3.6.1	Model	49
3.6.2	Anomalies	49
3.7	Interaction	50
3.8	Use cases	51
3.8.1	University	52
3.8.2	Industrial control system	52
3.9	Discussion and limitations	53
3.10	Conclusions and future work	54
4	Contextual Analysis of Anomalous Events	55
4.1	Understanding the Context of Network Traffic Alerts	56
4.2	Introduction	56
4.3	Related Work.	57
4.3.1	Alert Visualization	57
4.3.2	Exploration.	58
4.4	Problem statement	59
4.4.1	Data acquisition	60
4.5	CoNTA.	61
4.5.1	Exploration process.	61
4.5.2	Timetable	62
4.5.3	Context view	65
4.5.4	Conversation view	66
4.5.5	Attribute view	66
4.5.6	Classifier integration	67
4.6	Interaction	68
4.7	Use cases	69
4.7.1	Water plant.	69
4.7.2	University	71
4.8	Discussion and limitations	72
4.9	Conclusions and future work	73
5	Exploration of Multivariate Sequences	75
5.1	Exploring Event Sequences using Rules, Aggregations, and Selections	76
5.2	Introduction	76
5.3	Related Work.	78
5.3.1	Event Overview	78

5.3.2	Event Searching	78
5.3.3	Event Exploration	79
5.4	Exploration.	79
5.5	Rules.	81
5.5.1	Formal theory	81
5.5.2	Rule visualization.	82
5.5.3	Rule interaction	84
5.6	Pattern Aggregation	85
5.6.1	Structural sequence overlap.	85
5.6.2	Sequential sequence overlap	86
5.7	Selections	87
5.7.1	Context	88
5.7.2	Attributes	88
5.7.3	Interaction	88
5.8	Use cases	89
5.8.1	VoIP traffic.	89
5.8.2	Hospital records	92
5.9	Discussion and Limitations	96
5.10	Conclusion and Future Work	97
6	Hypothesis Testing & Generation in Wildlife Traffic	101
6.1	Hypothesis testing in Lekagul sensor Traffic.	102
6.2	Introduction	102
6.3	Data	103
6.4	Exploration.	104
6.5	Q1: Daily patterns	105
6.6	Q2: Periodical patterns	110
6.7	Q3: Unusual Patterns.	113
6.8	Conclusions	117
7	Rapid Reverse Engineering of Malware Behavior	119
7.1	Rapid Malware Analysis and Reverse Engineering.	120
7.2	Introduction	120
7.3	Related Work.	121
7.3.1	Malware discovery	121
7.3.2	Malware Identification	122
7.3.3	Deep Packet Inspection.	123
7.4	Eventpad	123
7.5	Use cases	125
7.5.1	Problem statement	126
7.5.2	Experimental setup.	128
7.5.3	Partition Strategies	128
7.5.4	Ransomware	129
7.5.5	University Traffic	133
7.6	Discussion	135
7.7	Conclusions and Future work.	136

8	Conclusions	137
8.1	Overview	138
8.2	Reflections	139
8.2.1	System components & Integration	139
8.2.2	Eventpad: Interpretability in Security Visualization?.	141
8.2.3	A hybrid model for Network Intrusion Detection	142
8.2.4	Automatic support: overview first, explanations on demand?	143
8.2.5	Security starts with understanding	144
8.2.6	Recommendations for building security tools	145
8.3	Future work	146
8.4	In Conclusion	148
	References	149
	Summary	177
	Samenvatting	179
	Curriculum Vitæ	181

Preface

I remember the day well when I entered Jack's office four years ago. As a Master student in Software Engineering I was looking for a supervisor in visualization to support me in the visual analysis of a complex parsing algorithm. After showing my interest in research and results from previous research programs (e.g., TU/e Honor's program), Jack gave me one option: "I will become your supervisor if you do a PhD in our group". Surprised by the spontaneous reaction I knew that it was going to be an exciting and yet a very valuable experience for me to become part of this group. I guess that was settled then :).

First of all, I want to thank my promotor, prof. dr. ir. Jarke J. van Wijk. Jack, it has been an honor for me to be part of the visualization group and to have you as my mentor. Your feedback has always been spot-on, constructive, and practical. As a non-graduate in visualization I had a lot of things to learn with respect to design and user-interaction. We sure had to laugh a lot during our elaborate discussions or my clumsy prototype ideas. Thank you for giving me the freedom to get in touch with industry and to steer the research to their challenges accordingly.

Furthermore, I would also like to thank my co-promotor prof. dr. Sandro Etalle. Sandro, thank you for sharing your ideas and enthusiasm in both security and entrepreneurship. Your advice and honesty both as co-promotor and business coach have been very valuable to me. Jack and Sandro, thank you for being my mentors in both academia and industry and making me part of such a tight yet very skilled group of people!

I would like to thank the SpySpot members Jerry den Hartog, Ömer Yuksel, Davide Fauri, and Allard Dijk for their help throughout the project. I really enjoyed working in such a multidisciplinary team. Ömer, thank you for supporting me during the start-up of the PhD and assisting me on the machine learning side of the project.

I also would like to thank the external partners in this project, consisting of Erik Boertjes and Sander de Kievit from TNO; Jan-Kees Buenen and dr. Danny Holten from SynerScope; and dr. Damiano Bolzoni and dr. Elisa Costante from SecurityMatters. Erik and Sander, thanks for your interest in the project, your intermediate feedback, and the enjoyable progress meetings. Elisa, thanks for all your efforts and commitment to the project. Your feedback during the prototype testing was indispensable. I also thank Synerscope for being part of the project. A special thanks goes to dr. Luca Allodi for his guidance during the preparation of the Black Hat submission.

I thank prof. dr. Mykola Pechenizkiy (Eindhoven University of Technology), dr. Michel Westenberg (Eindhoven University of Technology), prof. dr. ing. Jörn Kohlhammer (Technische Universität Darmstadt), dr. Lane Harrison (Worcester Polytechnic Institute Massachusetts), and prof. dr. Alex Telea (Rijksuniversiteit Groningen) for accepting the invitation to join the committee and providing me valuable feedback throughout my PhD career.

You have always supported my work and I enjoyed the time we spent together at the conferences. prof. dr. Mark G.J. van den Brand, thank you for being my supervisor during the Master program and for being the chairman in this committee.

Every project has its ups and downs. In this project it was very difficult to obtain real-world network traffic samples. This project would not have been possible without the help of the companies SecurityMatters, Motto Communications, and the Ministry of Defense. I especially would like to thank Jeroen Teeuwen and Kay Reijnen from the external company Motto Communications for their support and trust in me during the internship. Your critical view towards existing security solutions and curiosity for new suggestions are an example for other companies. I also would like to thank Nathalie Lokhorst and CISO Jaya Baloo from KPN for their support and making us members of their Computer Emergency Response Team. Nathalie en Jaya, bedankt voor jullie steun, enthousiasme en vertrouwen in ons onderzoek!

My PhD experience would not have been the same without the visualization group: Kasper Dinkla, Stef van den Elzen, Roeland Scheepens, Alberto Corvò, Ji Qi, Humberto García Caballero, Dennis Dingen, Dennis Collaris, Martijn van Dortmont, Mickael Verschoor, Huub van de Wetering, Andrei Jalba, Meivan Cheng, Michel Westenberg, and Robert van Liere. I have always enjoyed working in the group. Besides the in-depth technical discussions there was also room for chats and laughter. Stef, Kasper, Roeland, thanks for sharing your experience during the PhDs and for still joining the VIS events. Paul, Dennis, Alberto, Humberto, I really enjoyed spending time with you in the office and during outdoor events. Thanks for your help and joining me in this adventure. As the first organizer of the VIS Games, I am proud of the group and I hope we can continue organizing such events (including the PhD dinners) in the future.

Additionally, I would also like to thank my other colleagues, Renata Raidou, Daniela Modena, Marta Regis, Maximilian Konzack, Quirijn Bouts, Thom Castermans, Arthur van Goethem, Bart Jansen, Aleksandar Markovic, Wouter Meulemans, Sander de Putter, Roy op het Veld, Anne Postma, Hendrik Strobelt, Eamonn Maguire, Roman Pyzh, Fritz Lekschas, Siming Chen, Xixi Lu, Bart Hompes, Joos Buijs, Alok Dixit, Tim Ophelders, Bettina Speckman, Stefan Thaler, and Kevin Verbeek for the nice conversations and fun times we had during events and conferences. Outside academia, I thank Emma Bindels, Joep Roebroek, and Nick Peeters for the enjoyable and adventurous game nights.

The PhD has been an invaluable experience for me, but I also believe that this is not where it ends. The bridge between academia and industry is large and in order for research to become truly valuable for industry, it has to be scaled and tailored to customer needs. I would like to thank Josh Mengerink, Dennis Cappers, Joey van de Pasch, and Kay Reijnen for supporting me in this quest. I think we have an amazing team and I am looking forward to work with you on our projects. Besides the team I would like to thank Harold Weffers, Steven van Huiden, Jan Hubers, dr. Robert van der Drift, Tom Lemmens, and Barry van der Meer for their support and advice during the pre-startup phase of the company.

This section is dedicated to three people in particular, with whom I spent most my time in and outside my academic career: Josh Mengerink, Bram van der Sanden, and Maikel Leemans. Who would have thought that after nine years we would be still running around

at the university and finishing our PhDs simultaneously. I have never met a group that is as skilled, motivated, and inspiring as you guys. No matter how bad or large the obstacles were, we have always managed to stick together and supported each other in both good and bad times. Together with Wouter van Heeswijk, Sanne Wouda, Sander Jurgens, and Kris van Tienhoven we have formed a great team that is worth remembering. I thank you all for being part in this journey. I hope we can keep sharing these milestones in the future ;).

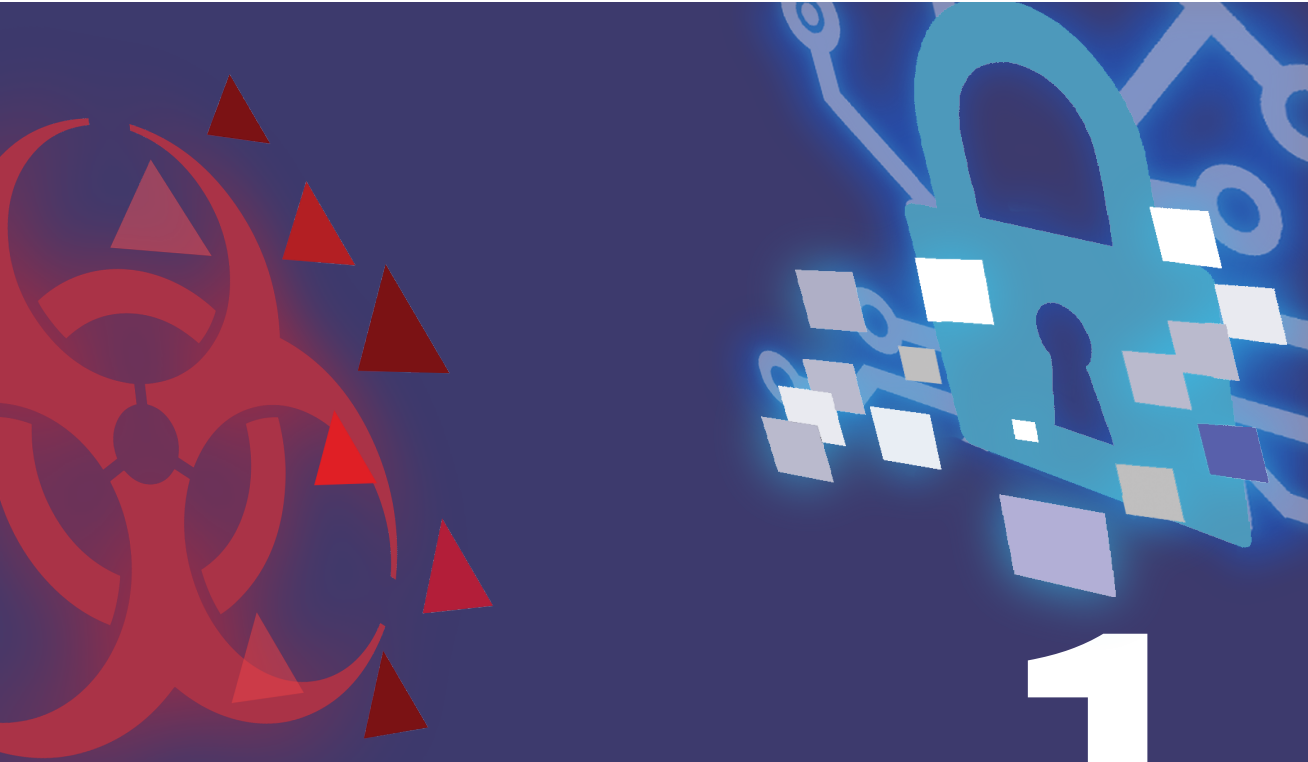
Many thanks goes to John Niemarkt, Pascal Wesolek, Paul Roozen, and the members of the dive club Nemo for introducing me to the sport and showing me the relaxing world below the water surface. Without you, I would have probably lost my mind way earlier :p. I am proud to be part of such a enthusiastic and tight group. I also would like to thank the TiMeS Lan crew for the fun times we had during the organization of our annual events.

I could not have done this project without the support of my friends and family. In particular Dennis, Rob & Karin, Suzanne & Lars & Niels, and Nancy & Herman. Nancy & Herman, bedankt voor al jullie steun en vertrouwen tijdens de studie en het promotietraject. Suzanne & Lars & Niels, bedankt voor alle tijd en vooral gezelligheid. Boys, bedankt voor alle steun door dik en dun. Zonder jullie was ik waarschijnlijk veel eerder klaar geweest :p. Dennis, pap, mam, woorden schieten tekort voor wat jullie voor mij hebben betekend. Dank jullie wel om dit mede mogelijk te maken. Bedankt voor alle steun, gezelligheid en goede zorgen.

Finally, I would like to mention Jos & José Gorissen, my grandparents Cisca & Jan Cappers, and Riet & Henk Jooren. To those who followed my career every step of the way, but unfortunately are unable to witness this milestone. For all of those I have failed to mention:



*Bram Cappers
Eindhoven, September 2018*



Introduction

1

1.1. Motivation

A nuclear facility disabled in Iran [193]; 6,5 million user credentials stolen from LinkedIn [332]; network environments infiltrated via vulnerable Cisco routers [302]. These are just a few examples of headlines that occur almost daily on international television and newspapers. Cyber-attacks have grown in number and sophistication making cybercrime currently the number one threat in our ICT society [75, 102, 151, 321, 350].

Cybercrime has changed dramatically in the last decade [206, 233]. Computer viruses are no longer designed to target everybody whenever possible, but are often tailor-made to strike once and cause severe damage. Especially for critical infrastructures (such as power plants), hackers nowadays are willing to design complex attacks to maximize the damage in such facilities. These targeted attacks (e.g., Advanced Persistent Threats - APTs [280]) use background knowledge to hide their communication inside a computer network, which makes the discovery of such attacks using high-level traffic statistics difficult. Inspection of low-level communication content enables us to detect APTs [160] in deeper layers of the network, but the heterogeneity and volume of the communication makes the discovery of attacks in this data a challenge.

Besides security, the analysis of network communication also plays an important role in system analysis [54]. Software and machines nowadays are composed of hundreds of sub-systems interacting with each other to perform complex tasks, such as making computer chips or providing worldwide telecommunication services. The discovery of for instance suboptimal resource usage or bottlenecks can help companies to simplify and improve systems in the future. We believe that the protection and analysis of complex systems starts with understanding the underlying network [92], but challenges remain:

- How can we *detect* these viruses if we do not know what we are looking for? and
- How can we *explain* what is happening in our increasingly complex systems?

In this dissertation we study how we can increase the understanding of networks by obtaining insights from visual exploration of the network communication. In particular, we explore how we can use visualization techniques and domain knowledge to assist machine learning in the detection of targeted attacks and improve exploration through user interaction.

1.2. Computer Networks

Networks play a central role in managing complexity. Software systems for example are decomposed into smaller components to make them better understandable, hospitals are divided into different departments to deal with diseases in specific body parts, and companies divide their work over multiple divisions to make fast delivery of services feasible. The result is a network where lots of information is exchanged between different components. Although decentralization reduces the complexity of systems, understanding the overall control flow between these components becomes more difficult. In Chapter 5, we show for

instance how hackers can manipulate communication between hardware components to hijack networks for their own greater good.

In practice, companies and engineers want to gain insight in the behavior of their networks for three purposes:

- *System optimization*: The behavior of complex systems in general is hard to comprehend. The identification of unexpected patterns or suboptimal resource usage can help companies to understand what is happening in their environments and provide improvements where necessary.
- *System debugging*: The integration of different systems is often a source of errors or undefined behavior. Analysis of system communication enable analysts to discover potential bottlenecks or redundant behavior.
- *System monitoring*: Complex systems often use valuable resources such as sensitive data or expensive hardware to provide services to their target customers. Illegitimate usage of these resources can seriously (physically) damage these assets [60]. Companies therefore want to continuously check whether particular behavior such as conflicts of interest [49], user impersonation [128], communication hijacking [158] etc. is not present in the system.

In order to achieve these goals, we need to understand what data is being transferred and what type of data should be allowed and disallowed. In case of system optimization we often do not know what we are looking for, whereas for the protection of systems obtaining an overview of all desired and undesired behavior is difficult. For both purposes, insights can be gained from visual exploration of the system's network traffic.

1.3. Multivariate Event Data

Many domains try to gain insight in their networks by logging events. Hospitals for instance store health records to analyze bottlenecks in patient treatments, credit card companies store financial transactions to discover fraud, and supermarkets log purchasing histories of customers to optimize their shop interior (Figure 1.1). What all these domains have in common is that they are interested in the discovery and understanding of *patterns* and *outliers* (also referred to as anomalies).

Besides the time of occurrence and type of event, in practice additional properties (also referred to as *metadata* or *attributes*) are stored describing how and under what circumstances the event was generated. Health records for instance can store information about the treatment of a patient, the name of the doctor, the department, how long the treatment took etc. We call events with metadata *multivariate* events.

The exploration of multivariate event logs is still a challenge due to their size and variety. Even for computer networks consisting of a few nodes, the amount of communication can be in the order of hundreds of thousands of events per second. In addition, the heterogeneity in system events can easily require the analysis of hundreds of attributes and more. This high-



Figure 1.1: Examples of multivariate networks and their communication: a) items bought in a supermarket; b) user click behavior on websites; c) packets sent between computers; d) patient treatments in a hospital are all examples of data that is being stored for the analysis and understanding of networks.

dimensional space makes exploration using just machine learning techniques difficult [114] as domain knowledge is required to focus on relevant aspects.

To cope with large volumes of data, current event analysis tools often work with *aggregated* data (also known as flow data [59]) to analyze high-level network properties, such as the number of events transferred per second and the size of the transferred data. However, for the detection of domain-specific virus attacks this level of abstraction may not give sufficient information (Section 2.6). Furthermore, the analysis of sequential patterns in event logs is typically limited to a single attribute at a time, thereby ignoring potential correlations that can exist between attributes. For root-cause analysis of malicious events both sequential properties and multivariate data should be explored simultaneously, since values in multivariate data are often crucial to understand patterns in sequences and vice versa.

1.4. The role of Visual Analytics

The analysis of system events plays an important role in the protection of cyberphysical systems. The sudden reboot or shutdown of a computer for instance can be an indicator that something is wrong in the network, but why do we need visualization if we have machine learning and artificial intelligence to detect intrusions automatically? Automatic methods have shown to be beneficial for the detection of both stealthy and brute-force attacks [114, 325]. However, to ensure that no attacks are missed, fully automated methods often tend to misclassify normal events as malicious. These are also referred to as *false positives* or *false alerts*.

Even when fully automated techniques claim to have false positive rates of less than 1%, in environments where thousands of events per second are generated, this can lead to tens of alerts per second. In modern System Information and Event Management (SIEM) frameworks such as AlienVault [7], system analysts have to deal with thousands of alerts per week and more (Figure 1.2). To remedy this, visualization could be an effective mean for the following reasons:

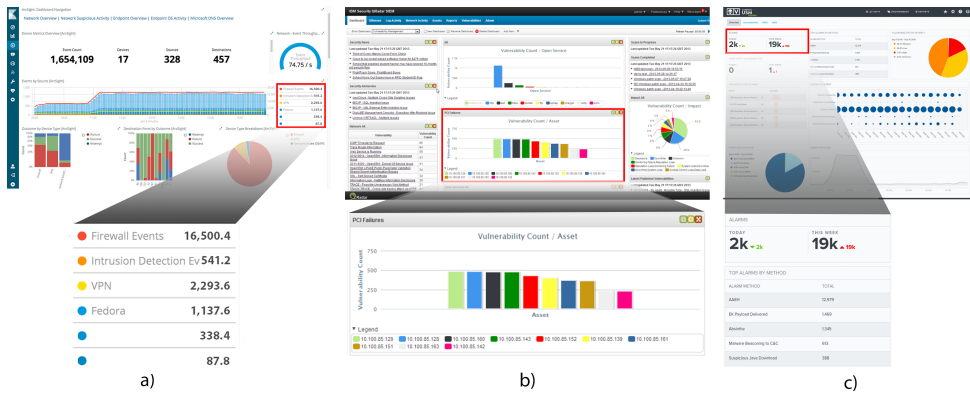


Figure 1.2: Fully automatic anomaly detection systems such as a) HP Arcsight [144], b) IBM QRadar [149], and c) AlienVault [7] still produce large amounts of false positives. How can we find the alerts that are relevant for the protection of our systems?



Exploit human cognition: The human brain excels in the discovery of patterns, outliers and distortions in visual input. The high-end feature extraction (e.g., positioning, orientation, color, shape) of the brain enables human to effortlessly detect and recognize objects in their environment. By presenting our data in a visual way we can exploit this processing power to obtain new insights from the data that might have gone unrecognized when solely relying on data-mining techniques or statistical methods.



Explain alerts: Events can only tell you what has happened in the environment, but not why they have happened. Human domain knowledge and other data sources can assist in assessing the severity of an alert by providing context. Knowing for instance that a machine was turned off for maintenance can trigger human cognition to relate the presence of alerts to these events.



Adjusting automated techniques: Concluding an alert to be false does not prevent it from happening again in the future. By incorporating a human in the decision-making process, we can let the user improve classification results by incrementally tweaking parameters of the algorithm or by applying the techniques on different parts of the data.

Even for networks consisting of a few nodes, the amounts of communication can be tremendous. A pure visualization approach is therefore often impractical. However, pure automated techniques often lack context and fall short as a result of large amounts of false alerts. Where automated techniques are fast in analyzing data but can be error-prone, humans are slow but more accurate. In this dissertation we therefore aim for a *visual-analytics* approach [166] where we combine algorithmic support, visualization, and human interaction to get the best of both worlds.

1.5. Objective

The goal of this dissertation is to combine anomaly detection [51], Deep Packet Inspection [253], and data visualization [48] to enable the detection of targeted attacks. As long as artificial intelligence cannot outperform human domain knowledge and cognition, we believe that humans play a vital role in the discovery and assessment of anomalies. In order to cope with the large data volumes, however, the use of automated techniques is unavoidable. To this end, the main question addressed in this dissertation is:

How can we use interactive visualization techniques and automated methods to discover relevant patterns and anomalies in large event collections?

The word *relevant* in the research question refers to the challenge of assessing the severity of an anomaly. Especially when dealing with automatically generated alert collections, distinguishing false positives from true alerts is a nontrivial task.

In order to answer this question, we aim for a hands-on experimental approach where we design end-solutions based on hypotheses and observations we made from both the data and the problem domain. Pretorius et al. [256] already showed that these domains may not necessarily coincide. Although the main application area of the dissertation is focused on network traffic analysis for cybersecurity, the developed techniques are not limited to this domain. For instance, in Chapters 5 and 6 we demonstrate how to use the techniques to enable the analysis of patient health records and vehicle travel patterns.

1.6. Outline & Contributions

In this dissertation we look at the research question from two different perspectives, namely:

- the types of data sources that are involved in the discovery of patterns and anomalies, and
- the types of patterns and anomalies that can be discovered.

From a data-perspective we can identify three sources of information, namely:

- multivariate event data describing what happened in the system,
- alert data indicating the severity of (collections of) events, and
- domain knowledge to assess the relevance of a particular observation.

We could consider alert data as an additional attribute in the multivariate event data. However, alert data can be defined at higher levels of abstraction beyond the scope of an event (e.g., at the level of a source, sequence, or a collection of events) that we can exploit using visualization.

In Chapter 2 we show that with respect to the detection of patterns and anomalies, we can identify three different classes, namely *point*, *contextual*, and *collective* anomalies. Figure 1.3

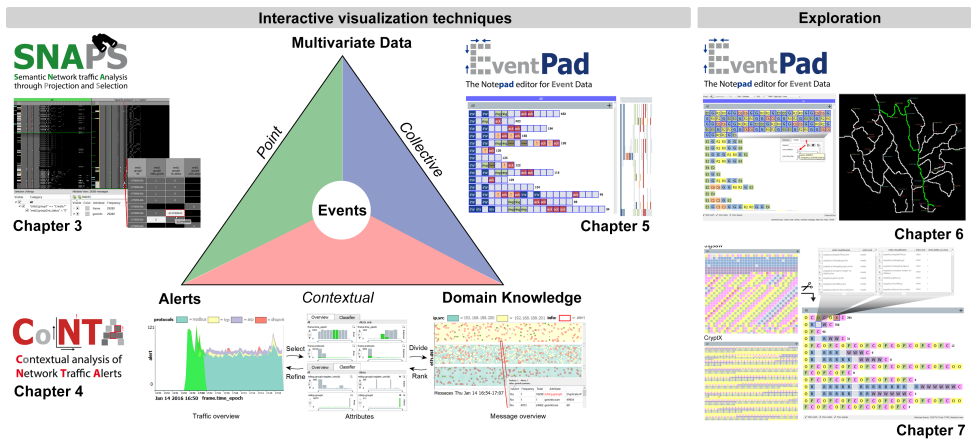


Figure 1.3: Overview of the chapters in this dissertation with respect to the research question. Every chapter addresses a different class of anomalies along with two or more data sources. Chapters 6 and 7 present case studies of the developed techniques on real-world data.

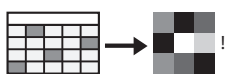
shows how each chapter contributes to the study of patterns and anomalies with respect to the different data sources and classes of anomalies. The proposed systems were designed to discover and analyze anomalies in event logs in different ways. The novelty of the proposed techniques therefore do not limit themselves to the cybersecurity domain. Although the use cases of the prototypes are mainly focused towards security problems, the systems are designed to work with any tabular data in general. This is also illustrated in Chapters 5 and 6. The dissertation is structured as follows.

Chapter 2 provides a background overview of interactive visualization techniques for event visualization and network traffic analysis. This chapter also presents an overview of different classes of patterns and anomalies along with a review of existing visualization tools for cybersecurity.

Chapters 3 to 5 present newly developed visual analytics systems for the discovery of different types of patterns and anomalies in multivariate event logs. In Chapter 6 and 7 we present different data explorations, where we use our techniques to discover illegal traffic activity in a wildlife preserve and analyze ransomware activity [219].

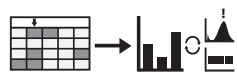
Finally, Chapter 8 concludes the dissertation by providing an overview of the results and techniques. In addition, we reflect on the results and provide guidelines on how to approach targeted attack analysis in network environments using visualization.

The key contributions of this dissertation are:



In Chapter 3 we present a novel exploration method to discover anomalous events by converting the event metadata to a pixel visualization. Combined with an online classifier, parts of the metadata are lit up when events contain values that are classified as malicious. Through interaction, users are enabled to explore the validity of the metadata attribute

space and refine classification results where necessary.



When trying to assess the relevance of an anomaly, context plays an important role. For instance, although the access of a file X does not have to be malicious in general, it can be considered dangerous when performed by a certain user. The way we split our data therefore determines the type of anomalies that stand out. In Chapter 4 we present a system to inspect alert data from different perspectives. We show how visualization and interaction can be used to enable analysts to discover high-level threats in a collection of low-level alert collections.



Chapter 5 focuses on the analysis of anomalies in event sequences. In this chapter we present a system called Eventpad that enables rapid and cost-effective discovery of patterns in event collections by visualizing them as blocks on a screen. Rules enable users to highlight and visual encode event properties that are of interest. Automated techniques such as clustering and alignment in turn can use this labeling to discover patterns between event sequences. Similar to a notepad editor, find & replace functionality and conditional formatting can be used to quickly search and highlight outliers in the data.



In Chapter 6 we present a case-study of Eventpad on the VAST Challenge 2017 Mini Challenge, where we apply anomaly detection on vehicle travel patterns in a Wildlife preserve. We show how we can combine contextual analysis and hypothesis testing with user interaction to enable rapid discovery of patterns without having the notion of an alert. This enables analysts to proactively search for anomalies according to their expectancy model, rather than trying to find an explanation of automatically discovered anomalies afterwards.



In Chapter 7 we extend the Eventpad system with temporal views to quickly study file access patterns in malware traffic. In particular, we look at the behavior of ransomware viruses that aim to deliberately block access to files on a user computer in exchange for money. Based on the discovered patterns we test if the malware samples are present in recorded samples of the university's office network.

Finally, Chapter 8 concludes the dissertation by providing an overview of the research results and reflecting on the lessons learned. We provide general guidelines towards the usage of automated techniques and visualization for the detection of targeted attacks in network environments along with directions for future work.

1.7. Publications

Publications in scientific conference proceedings and journals:

4. **B.C.M. Cappers and J.J. van Wijk**, Semantic Network Traffic Analysis through Projection and Selection, *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)*, (2015), pages 1-8 (Chapter 3).
3. **B.C.M. Cappers and J.J. van Wijk**, Understanding the Context of Network Traffic Alerts, *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)*, (2016), pages 1-8 (Chapter 4).
2. **B.C.M. Cappers and J.J. van Wijk**, Exploring Multivariate Event Sequences using Rules, Aggregations, and Selections, *IEEE Transactions on Visualization and Computer Graphics*, (2018), 24, 1, 532-541 (Chapter 5).
1. **B.C.M. Cappers, P.N. Meessen, S. Etalle, and J.J. van Wijk**, Eventpad: Malware Detection and Reverse Engineering using Visual Analytics, *To be published in the IEEE Symposium on Visualization for Cyber Security (VizSec)*, (2018) (Chapter 7).

Publications as a result of data challenges and research events

2. **B.C.M. Cappers and J.J. van Wijk**, Eventpad: Rapid Event Mining with Visual Analytics, *Proceedings of ICT.Open*, (2018) (**ICT.Open Best Demo Award 2018**).
1. **B.C.M. Cappers**, Exploring Lekagul Sensor Events using Rules, Aggregations, and Selections, *Visual Analytics Science and Technology (VAST) Challenge*, (2017) (**Visualization award “Elegant Tool for Hypothesis Testing and Generation”**) (**Best Poster Award 2017**) (Chapter 6). (Chapter 6)

Publications in Industrial reports, magazines, and conferences:

2. **B.C.M. Cappers, J.J. van Wijk, and S. Etalle**, Eventpad: A Visual Analytics approach to Network Intrusion Detection and Reverse Engineering, *KPN Cyber Security Perspectives*, (2018), pages 62-65.
1. **B.C.M. Cappers**, Eventpad: Rapid and Cost Effective Malware Analysis using Visual Analytics, *Black Hat Arsenal USA* (2018).

Publications to which I contributed during my PhD, but that are not included in the thesis:

2. **J.G.M. Mengerink, B. van der Sanden, B.C.M. Cappers, A. Serebrenik, R.R.H. Schiffelers, and M.G.J. van den Brand**, Exploring DSL evolutionary patterns in practice: a study of DSL evolution in a large-scale industrial DSL repository, *Proceedings of the International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, (2018).
1. **J. Qi, C. Liu, B.C.M. Cappers, and H.M.M. van de Wetering**, Visual Analysis of Parallel Interval Events, *to be published at EuroVis*, (2018) (**EuroVis Best Short Paper Award 2018**).



Background

The research question of this dissertation is multi-disciplinary, covering different topics in the area of data visualization, security, anomaly detection, and data science in general. In order to give our work context, we provide an overview of the current techniques and remaining challenges in each field with respect to the detection of anomalies for cybersecurity.

2.1. Data science

Data are very rich. Data can tell you what has happened, how systems and phenomena work, where they can be improved or enable users to make predictions about future phenomena. However, it is difficult in general to get clear takeaways by solely looking at the raw data. Data processing, classification, and linking between multiple sources are often required turn *data* into *information*.

The goal of data science [77] is to extract knowledge or insights from data to understand and analyze actual phenomena. This includes the processes [53] of

- *data extraction*: obtaining the desired data to enable analysis;
- *data transformation*: preparing the data to enable the use of analysis techniques (e.g., machine learning, process mining, statistics);
- *data analysis*: application of statistical methods, automated techniques, and visual inspection to discover new findings in the data;
- *data presentation*: (visually) communicating results to target domains to enable decision-making; and
- *data monitoring*: testing the performance and validity of applied extraction, transformation, and analysis techniques.

Data analysis techniques are often combined as they all have pros and cons. Machine learning approaches, such as classifiers, can handle large amounts of data, but can be error-prone without proper tuning of parameters. Statistical methods can measure and compare complex data phenomena using metrics, but can provide a skewed [145] or incomplete [132] view of the system due to the lack of overview. Data visualization enables users to gain quick insights in the data, but becomes challenging when analyzing large amounts of data in short periods in time. In Section 2.7 we discuss the possibilities and challenges with respect to the extraction of network events. Techniques for transforming, analyzing, and presenting data are presented in Chapters 3 through 5.

2.2. Data visualization

Card et al. [48] define visualization as the use of computer-supported, interactive, visual representations of data to amplify cognition. The aim of data visualization is to identify trends, patterns, and contexts that would otherwise go unrecognized in unstructured (e.g., text) or structured (e.g., tabular data) data.

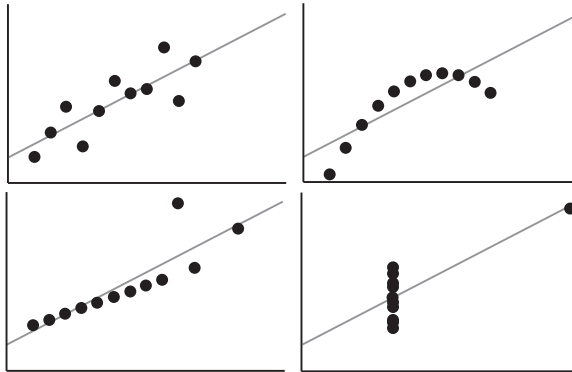


Figure 2.1: Ancombe's Quartet [132] Even when data sets share the same statistical properties, nontrivial distinctive patterns can be observed visually.

The human brain can recognize objects in just a few milliseconds by rapidly analyzing visual features such as color, orientation and positioning in parallel. In addition, the brain can also associate this visual input with ideas, hypotheses, and other sources of information to acquire new knowledge and in-depth insights. Data visualization tries to exploit these innate human capabilities by enabling the user to visually interact with the data.

Data visualization can be useful for data *exploration* as well as *explanation*. Steel et al. [150] refer to data exploration as trying to *find* the story that is behind the data, whereas data explanation aims towards trying to *tell* that story to somebody else. Data exploration is useful when dealing with lots of data without actually knowing what is in there. For such situations, visualization can assist users in discovering nontrivial insights [165]. In data explanation, visualization can help people to communicate and discuss results by deciding what and how to show the information to convey the desired message. This is also referred to as *data storytelling* [177].

If we know exactly what we are looking for and we know how to find our phenomena of interest, we would not need visualization. In this case, we can come up with an automatic method to solve our problem. However, if we don't know what we are looking for, visualization can give us valuable insights. Ancombe's quartet [132] for instance is an example where clear distinctions between data sets can be observed even though they are statistically identical (Figure 2.1).

In data exploration a question typically does not stand on its own. Once observations are made that deviate from our expectation, new subquestions emerge such as “why am I observing this?” and “could this be related to...?”. In order to enable this continuous reasoning and amplify cognition, interaction is key [48]. Yi et al. [347] identified different types of operations that can be used to interact with data in general:

- *Select*: mark something as interesting;
- *Explore*: show me something else;
- *Reconfigure*: show me a different arrangement;

- *Encode*: show me a different representation;
- *Abstract/Elaborate*: show me more or less detail;
- *Filter*: show me something conditionally; and
- *Connect*: show me related items.

This is typically achieved with interaction techniques such as *dynamic querying* [276], *linking* [39], *brushing* [166], and *semantic zooming* [247].

Data visualization can be subdivided into three communities [138], namely scientific visualization [235], information visualization [284], and visual analytics [166]. *Scientific visualization* focuses on visualization techniques for data representing physical phenomena (e.g., blood vessels [163], 3D molecules [243], volume rendering [86]), whereas the field of *information visualization* specializes in the design of interactive techniques to analyze abstract data for which no physical representation is given. Examples are for instance tabular, hierarchical, and time-series data.

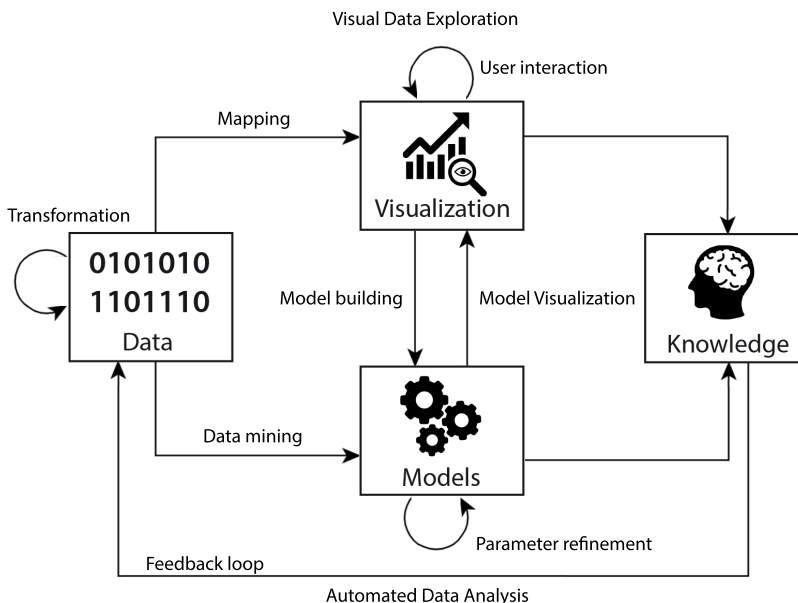


Figure 2.2: The model of Keim et al. [166] shows how visualization and models can be used to gain insights.

The field of visual analytics extends data visualization by combining analysis techniques with interactive visualizations for effective understanding of, reasoning about, and decision making on complex data sets [166]. This is illustrated in the model by Keim et al. in Figure 2.2. The introduction of algorithmic support in visual analytics is two-way. On one hand, the user is enabled to learn from the results that can be derived from the model. This can for instance lead to new questions or different perspectives on the data. On the other hand, the model can “learn” from the users by enabling them to adjust parameter settings or labeling the data with additional information. The latter is for instance popular in semi-supervised learning techniques [52, 360].

2.3. Users, Tasks, and Problems

The analysis of event collections enables users to investigate historic system events, to monitor system conditions, and to make predictions about future phenomena (e.g., a system breakdown) [84, 222]. Here are some examples of user categories that can benefit from interactive visualization of event collections.

2

Security Operations Center analysts

A Security Operations Center (SOC) is a facility where information systems such as data centers, servers, networks, desktops, and other endpoints are monitored, assessed, and defended. The main task of network engineers and security analysts in a SOC is to detect signs of possible cyber-attacks or network intrusions by monitoring these systems actively. A big challenge in these centers is to achieve overview and in-depth real-time inspection of the environment when dealing with tremendous volumes of streaming data.

Emergency Response Teams

Computer Emergency Response Teams (CERTs) [110] (also referred to as Computer Security Incident Response Teams) are groups of network and security experts that handle computer security incidents. Tasks of CERTs vary from ranking and escalating tasks and alerts [133] to coordinating and executing response strategies for the containment and remediation of IT threats. CERTs typically combine results of multiple network intrusion detectors and SIEMs to determine the severity of a threat. Big challenges here are how to deal with large false positive rates and how to combine information sources.

Digital forensics experts

Digital forensics [50] focuses on the investigation of cyber incidents after the attack has happened. The goal of digital forensics is to collect evidence in large recordings of network traffic, malware samples, hard disk drives etc. to identify people that are responsible for the attack and to develop countermeasures accordingly. Digital forensics can require months of analysis [159]. The challenge here is to enable efficient in-depth analysis of these large offline data collections [115].

Maintenance engineers

The maintenance of a critical infrastructure is a difficult task. The financial damage as a result of a system breakdown can be in the order of millions of euros per day, whereas preventive teardown of normal functioning machines increases the risk of material fatigue. The inspection of trends and anomalies in sensor data enable maintenance engineers to predict when machine parts need to be replaced. A big challenge here is how to relate low-level sensor events to the breakdown of specific machine components [222].

Network engineers


Network engineers are responsible for the deployment and management of the IT infrastructure inside companies. Their daily job consists of multiple tasks including the reception and processing of security tickets about odd network behavior [155]. Based on the intelligence that is obtained by CERT and SOC members they deploy updates in the network to solve vulnerabilities. This can vary from defining new firewall rules in the routers to re-installing or patching machines in the network. A big challenge for network engineers is how to com-


municate observations and tasks between different groups to implement security solutions as fast as possible.


2

2.4. Anomaly detection

Chandola et al. [51] define the field of anomaly detection (also known as outlier detection) as the identification of items, events or observations that do not conform to an expected pattern or other items in a dataset. Examples of anomalies are for instance data points not following a particular distribution (e.g., extrema), the occurrence of incorrect or infrequent values, or the repeated presence or absence of particular events. Anomalies can be defined according to different metrics [242]:

 *Distance:* the distance between data points is often used to determine anomalies. The assumption of this model is: the greater the distance between a point and the rest of the data, the more likely it will be different from the rest. The distance between points is defined by means of a distance metric, such as Manhattan distance [27], cosine similarity [287], and Euler distance [76].

 *Frequency:* data points can be considered anomalous if they occur too little (or too often) in the data set with respect to others. Some examples are the invocation of a deprecated function call, the (repeated) occurrence of incorrect or mistyped values, or the sudden change of a constant value over time.

 *Density:* the density model assumes that normal data points are close together and abnormalities are more isolated. This model is typically used in clustering techniques such as DB-SCAN [91] to deal with nonlinear manifolds. The icon shows an example where data points can be close to each with respect to distance, but can still be an outlier with respect to neighbor density.

Anomalies in general can be classified into three categories [51]. Figure 2.3 shows examples of each category for scatterplots and event sequences:

- *Point:* Point anomalies are events (or data points in general) that are anomalous with respect to the entire data set. An event B can be considered anomalous in an event log consisting of only A's if we define anomalies by frequency. Point anomalies are also referred to as *global* anomalies, since they are unusual irrespective of the context in which it was observed.
- *Contextual:* Contextual anomalies are data points that are anomalous with respect to a certain context. Consider for instance the event sequence in Figure 2.3b. An event sequence consisting of alternating A's and B's may not seem unusual, but it can be considered strange when the data is split by an attribute of choice (e.g., split by the user). Contextual anomalies are also referred to as *local* anomalies since they are only visible with respect to a subset of the data.
- *Collective:* Collective anomalies are collections of data points that together are considered anomalous. For example, events such as “Close gas valve” and “Light a fire” are not unusual in a combustion engine. However, the order in which the events occur

together can have severe consequences. Collective anomalies are related to the discovery of *patterns*. The Oxford Dictionary defines a pattern as “a regular and intelligible form or *sequence* discernible in the way in which something happens or is done.”

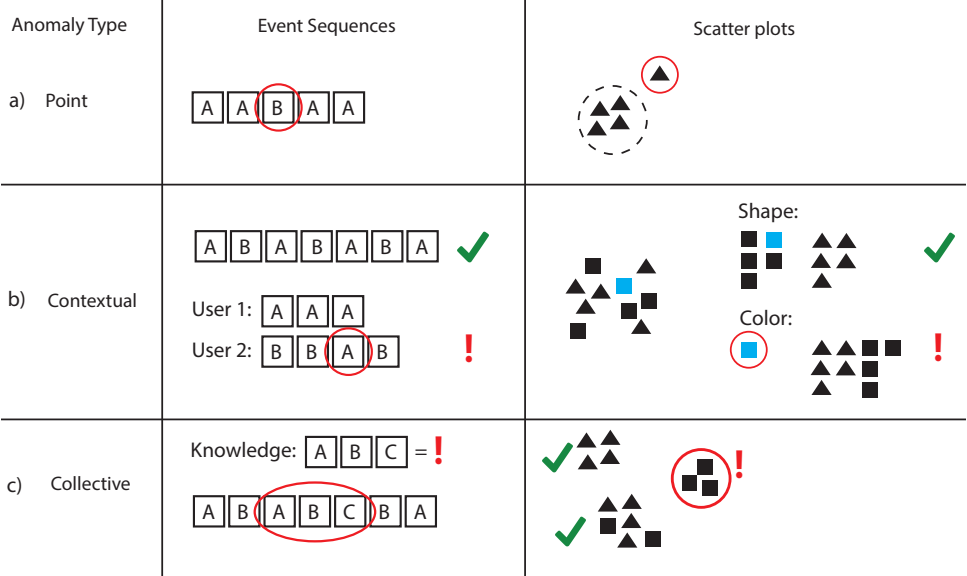


Figure 2.3: Three classes of anomalies: a) Point anomalies (unusual with respect to the entire data set) b) Contextual anomalies (unusual with respect to a subset), c) Collective anomalies (unusual as a group, not necessarily as individuals).

The detection and classification of anomalies is a nontrivial task and is closely related to the “Reference Class” problem [131] in statistics. Especially for contextual anomalies, the way we split our data determines the type of anomalies that stand out. The example in Figure 2.3b already sketches the problem when dealing with only two attributes such as shape and color. The number of possible perspectives unfortunately grows exponentially in the number of attributes. How should an algorithm (or an analyst) know which perspective of the data is more relevant than the other? In Chapters 3-5 we propose several techniques how we can use interaction to investigate contexts in an exploratory fashion.

Finally, anomaly detection can be either *time-aware* or *time-agnostic*. Time-aware anomaly detectors focus on the detection of anomalies in temporal data by taking time data into account (e.g., time between events, time of occurrence, event ordering). Time-agnostic anomaly detectors ignore temporal information and focus on the detection of anomalies on individual data points (e.g., inside the multivariate data of an event) or data aggregations.

2.5. Network Intrusion Detection

Anomaly detection in cybersecurity is crucial for the detection of undefined or malicious behavior. The surveys of Mitchells et al. [221] and Etalle [92] classify Intrusion Detection

Systems (in short IDSs) into three main categories, namely knowledge-based, behavioral-based, and specification-based intrusion detection.

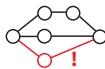


Knowledge-based systems detect intrusions by searching the data for known patterns of benign or misbehavior. Typical examples of knowledge-based detection techniques are for instance *black-listing* and *white-listing* approaches [98], where users define rules (e.g., signatures) specifying what an entity or system is (or is not) allowed to do. Knowledge-based detection is highly efficient to apply and easy to construct, but is unable to discover anomalies that are beyond the scope of these rules.



Behavioral-based anomaly detection discovers intrusions by assessing the severity of a new data point according to some baseline model. Behavioral-based detectors typically require a training phase to learn how regular activity in the system can be described.

Although behavioral-based detection has shown promising results in the literature [344], in practice these models are difficult to train on environments that are already compromised or change quickly over time [240]. As a consequence, the number of false positives can be high.



Specification-based anomaly detection is a special branch of anomaly detection that defines legitimate behavior of a system by means of a formal specification. Systems are considered intruded when their behavior deviates from this model. Ko et al. [173] showed that specification-based anomaly detection achieves lower false positive rates compared to behavioral-based anomaly detection by specifying policies for security-critical applications. Creating a full policy of a system in practice, however, is difficult and often time-consuming.

The boundary between knowledge-based and behavioral-based detection is not sharp. Heuristic detection [227] for instance enables analysts to specify rules that are softer compared to traditional signature-based approaches, but are often still bounded to an expectancy model the security analyst has of the environment. To improve distinguishability, Etalle [92] proposes a different categorization for these techniques based on whether they describe *acceptable* or *rejectable* behavior. Data visualization techniques typically aim towards behavioral-based intrusion detection because of their exploratory nature [162]. Even if we do not know what we are looking for, the visualization of historical information can assist in the detection of unknown attacks.

Finally, network intrusion detection can be done in an *online* or *offline* fashion. Online solutions are typically used in network monitoring applications [87] to detect intrusions by observing the network (near) real-time. This involves efficiently testing incoming data for particular signatures and measuring traffic behavior over time. Once a potential Indicator of Compromise (IOC) has been discovered, digital forensics [50] (also referred to as post-mortem analysis) analyzes the phenomena in greater detail by reverse engineering the behavior from recordings of large network samples.

According to intrusion detection surveys [114, 198], general-purpose state-of-the-art behavioral-based intrusion detection systems still suffer from large false positive rates. How-

ever, anomaly detection techniques by Yuksel et al. [349] and Costante et al. [70] show an impressive reduction in false positives when designing the techniques for specific domains, such as industrial control systems and database environments. In addition, both methodologies enable users to provide feedback on profiles that were learned by the anomaly detection model, illustrating the value of a human in the detection process. In the next chapters, we continue to explore the value of human feedback by combining intrusion detection models and user knowledge using visual analytics.

2.6. Advanced Persistent Threats

Until the early 2000s viruses were mostly designed to target large audiences, such as the consumer market [206, 233]. The virus is typically spread by email or via links on websites with one goal in mind: trying to infect as many as possible. Starting from 2005 [147] a different class of computer viruses emerged that, in contrast to traditional viruses, were specifically designed to target a particular infrastructure.

Once infiltrated (typically by means of social engineering [124]), the virus locates the system that it wants to target and stays under the radar by hiding its traffic in or alongside regular activity. These viruses either try to stay hidden and leak information to the outside world or aim to cause severe damage to the underlying infrastructure. This class of computer viruses are referred to as *Advanced Persistent Threats* (in short APTs) [280].

In 2010 a famous APT attack was executed on a nuclear facility in Natanz, Iran [60]. The Stuxnet [193] virus is held responsible for the destruction of up to 1,000 centrifuges by manipulating their rotor speed. Instead of shutting down the centrifuges (what would have resulted in an alarm), the virus altered the fan speed of the centrifuges by gradually increasing and decreasing it. This eventually resulted in a faster breakdown of the centrifuges. Other examples of more recent attacks and attempts are:

- 2015 Carbanak APT bank robbery [272];
- 2015 Four Month lasting attack on the New York Times [95];
- 2017 Attackers deploy new ICS Attack Framework [101];
- 2018 The (failed) hack attempt to a Saudi Arabian Petrochemical Plant [234]; and
- 2018 SmartInstall hack in Cisco Routers [302].

Although infiltration of targeted attacks in general is very difficult to prevent, we can detect signs of expansion and sabotage by analyzing the network traffic that is sent inside the environment. APTs typically try to hide their activity in the application layer of the network traffic. Deep Packet Inspection [253] enables us to analyze application data to keep track of undesired behavior in the network.

2.7. Network traffic

Data extraction plays an important role in cybersecurity. The type of data source (e.g., network traffic, twitter, newspapers), the location where the data is obtained, and the level of granularity determines the type of attacks that can be discovered. Counting events per second is quick and easy, but limits the discovery of attacks to bursts or drops in data volumes. Inspection of the full data enables the discovery of more complex attacks, but introduces challenges with respect to the analysis and recording of the data. In this section we discuss how network traffic is structured, how it can be obtained, and what the main challenges are with respect to the analysis of this data.

2.7.1. Non-intrusive vs. Intrusive Network Recording

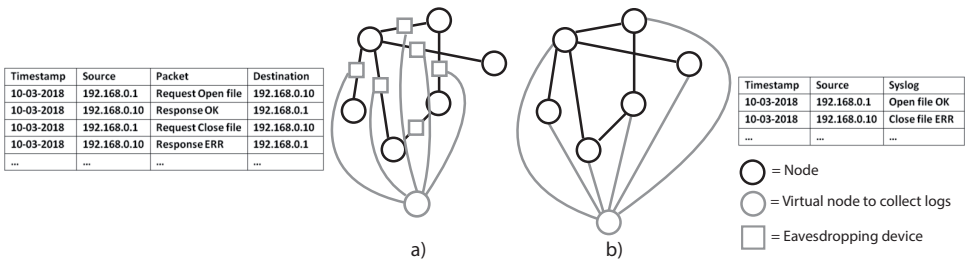


Figure 2.4: Different ways to extract event information from networks: a) Non-intrusive recording eavesdrops on the edges of the network. The network is unaware of the recording and events consist of source and target information. b) Intrusive recording requires the installation of event recording software on every node in the network. Events obtained from intrusive recordings contain source information.

In general, network communication can be recorded in two different ways, namely *non-intrusive* versus *intrusive* (also depicted in Figure 2.4). Non-intrusive data recording techniques collect data by “eavesdropping” on the edges of the network and the network is unaware of this recording. This approach is typically used in Network Intrusion Detection systems (also known as NIDSs) and traffic analyzers such as Wireshark [63] and Bro [244]. Intrusive data recording techniques extract events from the nodes in the network. This requires that nodes log all the actions they perform throughout system execution. Companies and organizations often perform intrusive data collection to study workflows in systems [54].

Event data obtained from non-intrusive data recordings typically have *source* and *destination/target* information (e.g., network packets). Intrusive event data are typically limited to source information only (e.g., system logs). Intrusion detection systems that operate on individual hosts or devices in the network are also referred to as Host-based Intrusion Detection Systems (HIDSs) [198, 255, 346].

Multivariate data collections can store properties for different aspects of the network, namely for nodes, events, and sequences (Figure 2.5). Examples are:

- *nodes*: source IP address, device name, time since last reboot, etc.;

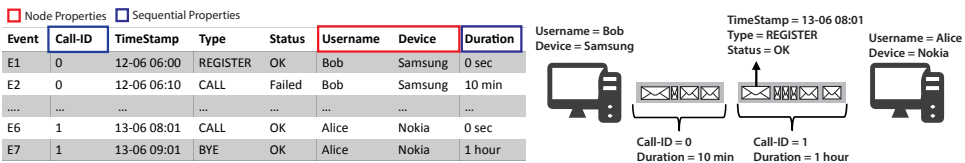


Figure 2.5: Node and sequential properties such as username, device name, call-id, and call duration can be encoded as event properties. The remaining attributes are examples of event properties.

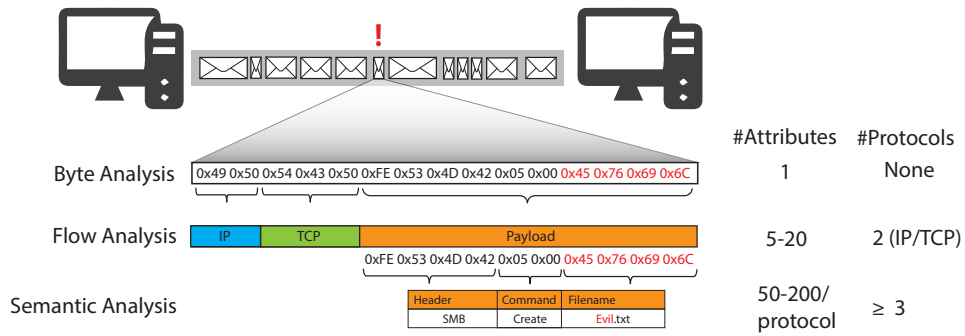


Figure 2.6: Network traffic can be analyzed at different levels of abstractions: a) byte-level; b) network-level; c) application-level.

- *events*: type of event (e.g., open file, close file), option parameters (e.g., read-only settings, file share properties), time of occurrence, etc.; and
- *sequences*: the number of events per sequence, sequence duration, session_id, etc.

In practice, node and sequential properties are often added as metadata to event records for the sake of self-containment. In case of node properties, this typically results in values that are constant over multiple events. Figure 2.5 shows examples of node, event, and sequential properties in Voice Over IP phone call records.

2.7.2. Deep Packet Inspection

Deep Packet Inspection (DPI) [253] is an overall term for data processing techniques that look in detail at the content of network traffic. Figure 2.6 shows different levels in which we can analyze traffic [241]:

- *Byte analysis* [65] analyzes network traffic by discovering patterns in a byte representation of the traffic. Any information related to the meaning of byte (sub)sequences is discarded. Statistical profiling [291] or n-gram analysis [130] can be used for instance to detect illegal shellcode sequences [6].
- *Flow analysis* [226] tries to gain better insight in the traffic by extracting values from byte sequences according to some *protocol* specification (also known as dissecting or parsing [4]). Network protocols describe how messages should be converted to byte

sequences and vice versa. Network packets are often composed of multiple nested protocols that each have their own purpose. Protocols such as IP and Ethernet are used to route packets to the desired destination, whereas protocols such as TCP [108] are used to ensure that packets are not lost during transfer. Application specific protocols such as Samba [268] or SIP [266] can be used to enable file access on network shares or enable the transfer of audio material (e.g., Voice Over IP [28]) respectively.

In case of flow analysis, the parsing of the network traffic is limited to the protocols that nearly every packet has in common, namely IP, Ethernet, and UDP [254] or TCP. The information obtained from these protocols enables analysts to study where, when, and how much traffic is sent over the network. However, the content of the messages (also referred to as *payload*) remains unknown.

- *Semantic analysis* [73] (or Deep Packet Inspection) enables the analysis of network packet content by also parsing application specific protocol data in the network traffic. This enables analysts not only to see how large a packet is, but also what it represents (e.g., access to a file, initiation of a phone call).

Depending on the type of packet (that is, the protocols it uses) specific attributes and values can be present. In addition, the number of possible protocol fields and options in application-specific protocols such as Samba is in the order of hundreds and more. This raises questions about deciding which protocol fields to analyze and how they can contribute to the detection of complex network attacks.

DPI data is typically stored in PCAP files [296]. Figure 2.7 shows an example of a dissected network packet representing a request to open a file called `srvsvc`. The packet contains data of 5 different protocols, the outer `Frame` layer is constructed by Wireshark. The last layer `SMB2` (Samba) stores information that is related to the file operation. This is also referred to as application data or *Layer 7* traffic [34]. The `Netbios` layer stores the size of the packet without network layers. This way the Samba protocol parser knows when the end of the packet has been reached.

In practice, Deep Packet Inspection is computationally intensive and often requires expensive hardware to enable real-time large-scale network monitoring [198]. Instead, current systems typically focus on flow analysis by recording Cisco Netflow data [59]. Burst and drop activity in the network can be an important indicator of compromise in network activity, but may not be a sufficient indicator for the detection of stealthy attacks at application-level. Chapter 7 for instance shows how Ransomware malware can become hard to detect when hiding file access activity in deeper layers of the network data and spreading the attack over a larger period in time.

Encryption

The techniques described in this dissertation with respect to the analysis of network traffic assume that the traffic is unencrypted. This either implies that the observed network traffic is not encrypted at all or the network capture system is able to decrypt and re-encrypt the traffic after inspection. Although insights can also be gained by analyzing unencrypted metadata

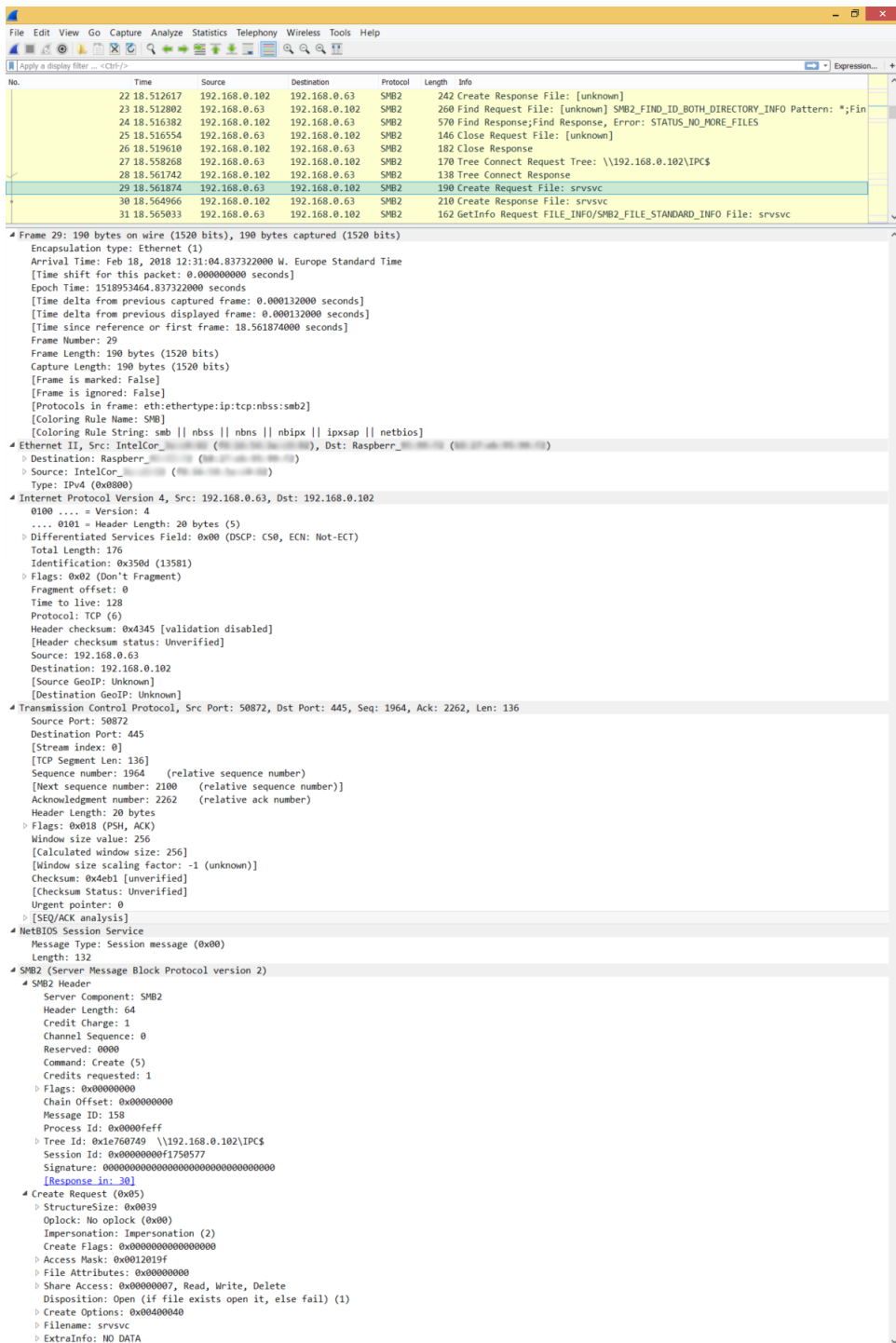


Figure 2.7: Example of one full network packet dissection. The network packet represents an open request to a file `srsvsc` located on a network share.

of security protocols such as Secure Sockets Layer (SSL) [140] and Transport Layer Security (TLS) [8, 78], Deep Packet Inspection becomes less effective.

Encryption of network traffic however does not imply safety. For critical infrastructures such as Industrial Control Systems, Fauri et al. [96] argues that encryption not only decreases data visibility for potential attackers, but also limits security tools in the detection of intrusions. By naively encrypting the network traffic in environments, attackers won't be able to read the content of the traffic, but also defenders won't be able to see the attacker once he has entered the system.

Privacy

Deep Packet Inspection in network monitoring is often questioned for privacy issues [10, 23, 112]. The Dutch telecom provider KPN [303] for instance was recently negatively in the news for admitting the use of DPI techniques for optimization of their infrastructure.

A balanced decision has to be made between protection by inspecting data versus preserving privacy by hiding it [81]. On one hand, people want to be safe from attacks on nuclear facilities and agree that network traffic should be monitored for the greater good. At the same time, users don't want their own data to be monitored as there might be a risk for the data to be misused or leaked to the outside world.

In general, systems that capture byte data from network environments are capable to perform Deep Packet Inspection. In this dissertation we show how we can use open source software such as WireShark [63] to gain deeper insights in recorded network samples. However, the application of DPI in large networks in practice can be difficult to achieve for several reasons:

- *Traffic volume*: Enabling DPI in network environments can be costly both in computation power and hardware. A single instance of a dissector such as Wireshark typically limits the analysis to a few thousand packets per second. Brown et al. [35] show that the capturing of raw traffic on a single instance is limited to just a few Gigabits per second. Enabling large-scale DPI requires a more complex architecture of dissectors and advanced storage management to keep up with the pace of larger network streams.
- *Data fragmentation*: Network files and messages are typically divided into chunks to enable transmission. In order to obtain a full insight in traffic content, packets have to be reassembled after which the parsing can start. Phenomena such as packet omissions, reordering, and retransmissions as a result of packet loss can make the reconstruction a tedious and nontrivial task.
- *Level of abstraction*: Network protocols consist of many options and technical details that are difficult to grasp to the untrained user. In-depth knowledge about the Internet Protocol Stack [257] and domain-specific protocols are required to extract information from this data.

Similar to European GDPR guidelines [30], the purpose and application of DPI should be transparent and should be built on a certain level of trust. Extraction of the full traffic content, however, is not necessarily required to increase awareness in network environments.

For example, in Chapter 7 we show how we can discover patterns in access behavior of machines by analysis of the Samba metadata in the traffic. Although this enables analysts to see when, where, and how files are accessed in the network, the content of the files was not considered.

2.8. Security visualization

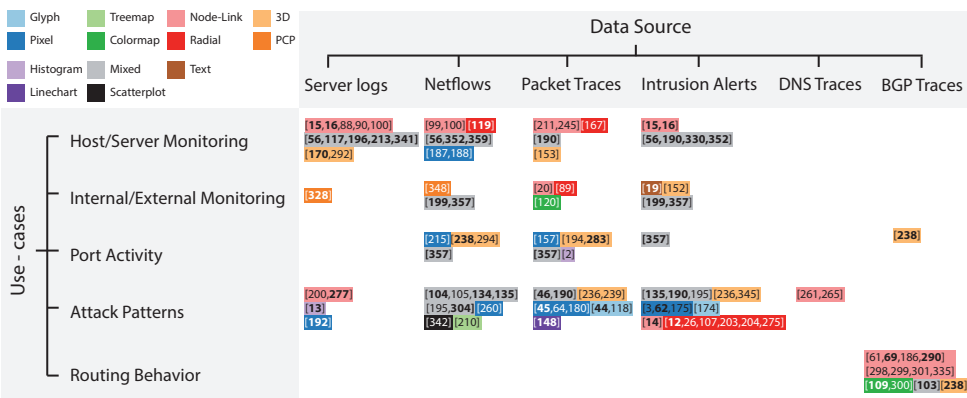


Figure 2.8: Extension of the security visualization taxonomy by Shiravi et al. [274]. The colors show the type of visualization that was mainly used in the papers. Labels in bold are papers that have been added to the survey as part of the extension.

Over the years many different visualization techniques have been proposed to assist security analysts in gaining insight in event logs and network traffic. Several survey papers have been written to categorize the techniques along different dimensions [17, 97, 125, 162, 169, 201, 274, 353, 354].

Shiravi et al. [274] for instance categorize visualization systems based on their input data source (e.g., packet traces, intrusion alerts) and use case (e.g., host monitoring, attack pattern detection), whereas a more recent survey by Zhang et al. [353] organizes the techniques based on their data properties (e.g., tabular, hierarchical) and visualization tasks (e.g., detection, correlation). Another survey by Dasgupta et al. [74] checks which visualizations are suitable for streaming data analytics or static data analysis.

To get a more complete overview of current security visualization solutions, we extended Shiravi’s survey [274] with the overview of Zhang et al. [353] and current state-of-the-art techniques in network security visualization. These were searched on the web and in the latest proceedings and journals of IEEE Symposium on Visualization for Cyber Security (VizSec), IEEE InfoVis and VAST (2016-2018). This was done by searching on the following keywords: “anomaly detection”, “security visualization”, “situational awareness”, “network traffic analysis”, “computer network logs”, “security event/system logs”, “alert visualization”, “intrusion detection”, “misuse detection”, and “attack pattern visualization”.

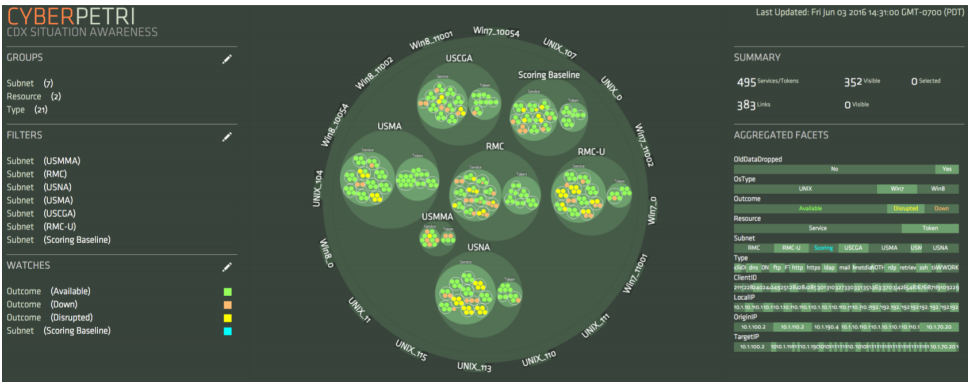


Figure 2.9: CyberPetri [15] visualizes the status of hosts by projecting log entries in the network hierarchy.

Figure 2.8 shows an extension of Shiravi’s survey where papers are color-coded according to their visualization technique. The citations in bold are additions to the original survey. The figure shows that a wide variety of visualization techniques has been used over the years. In the period 2004-2007 there was a trend to try out different visualization techniques by proposing new schemes to visual encode network data. More recent papers [134, 352], however, no longer focus on a single type of visualization, but combine multiple techniques together in a multi-view system (as indicated by the “mixed” category).

Every survey provides a different viewpoint on the domain, but most consider data source to be relevant for the categorization. Shiravi [274] for instance distinguishes between the analysis of Netflow records, packet traces, and server logs. On a higher level of abstraction, these can all be viewed as instances of event collections. Visualization of events in general has been studied extensively. In the following sections we propose two taxonomies for security and event visualization that enables us to bridge the gap between the two domains.

2.9. Taxonomy security visualization

In this taxonomy we categorize the visualization of security logs based on the type of events they analyze, whether they focus on the visualization of alerts, and how these alerts are established.

2.9.1. Host vs. Network events

Events can be recorded at the level of nodes or edges in the network (Section 2.7.1). Network-level security visualizations show network events, obtained from non-intrusive recordings, by explicitly visual encoding *source* and *destination* information in their solution. Visualizations such as NVisionIP [188] and TNV [120] for instance use a matrix visualization to display IP information.

Host-level security visualizations focus on the visualization of intrusive network traffic recordings. Petridish [15] (Figure 2.9) for instance analyzes server logs by generating an overview of the underlying network hierarchy. System logs and IDS alerts are visually encoded in the hierarchy depending on the location of the host machine where they originate. The QCat [328] system analyzes user downloads and login records by visualizing correlations inside the multivariate data using a Parallel Coordinate Plot (PCP) [331]. In addition, users are enabled to discover new anomalies by interactively defining and adding new axes to the PCP plot.

Other security systems do not make assumptions about the event data to visualize. These systems often offer the users the flexibility to let them decide on how to visual encode their data. Commercial Security Information and Event Management software such as SolarWinds [281], IBM QRadar [149], HP ArcSight [144], and AlienVault [7] fall in this category. Also visualizations by Keim et al. [167] and Hao et al.[134] enable users to decide which attributes to visualize and how they should be represented. These systems can visualize data in a flexible way, but Healey et al. [139] indicated that they provide little user guidance in effectively discovering areas of interest. For these systems in-depth visualization experience of the user is required to discover nontrivial insights.

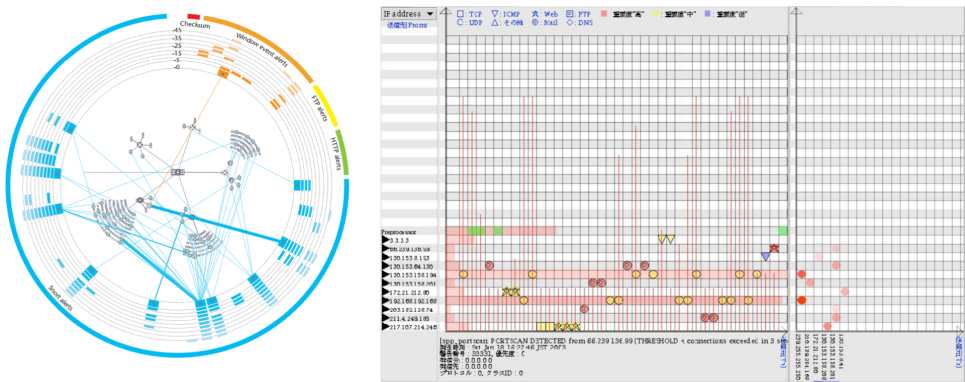


Figure 2.10: Examples of alert-based network security visualization. Livnat et al. [203] (left image) visualize network traffic alerts by encoding these based on what, when, and where they happened in the network. SnortView [174] (right image) shows sequential patterns in Snort alerts by encoding the alerts using glyphs.

2.9.2. Alert vs. Non-alert

Security visualizations often focus on the visualization of alert collections that are generated by some (third-party) Intrusion Detection System (IDS). For large alert collections enumerating the alerts in for instance a tabular view does not give a sufficient overview of the system.

IDS Rainstorm [3] for instance visualizes Stealthwatch [191] alerts by positioning the alerts in a pixel visualization according to their IP information. The color of the pixels indicates the severity of alerts over time. Livnat et al. [203] visualize network alerts by showing *what*,

when, and *where* they have happened in the network. The result is a radial display with a network topology in the center and different alert types defined in the outer rings of the visualization (Figure 2.10).

2

General-purpose systems often do not explicitly use the notion of an alert. Systems such as Tudumi [292], Girardin et al. [118], Rumint [66], and VISUAL [245] support the discovery of erroneous behavior by visual encoding of the data in a technique of choice and aim to present unusual patterns in the resulting image. Other systems, such as APT-Hunter [277], support the detection of malicious logins in event recordings by enabling users to define and search for login patterns using a query language.

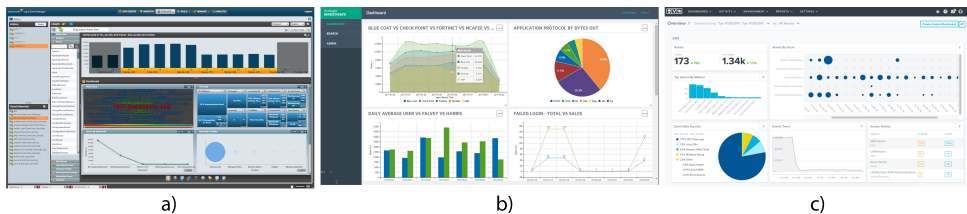


Figure 2.11: Current Security Information and Event Management Systems (SIEMs) often provide flexibility to generate high-level overviews of the data, but provide little guidance throughout exploration. Analysts require in-depth knowledge about the data and construction of visualization to discover relevant patterns.

2.9.3. Internal vs. External

Depending on the type of security system, alerts either originate from third-party applications or are generated according to some internal model. External alert collections are often precomputed on a data set and are considered as another input source for the visualization. SnortView [174] for instance tries to discover attack patterns by encoding Snort alerts as glyphs, positioned based on their time of occurrence. Daedalus-Viz [152] provides a real-time visualization for Deadalus alert collections by visualizing them in a 3D environment.

Internal alert visualizations focus on the visualization of alert collections that are generated according to some model or live classification algorithm. Landstorfer et al. [192] visualize event log records as a stack of pixels, where every pixel denotes the frequency of a value in that record. By means of filtering, uninteresting records can be removed from the data, after which the classifier is reapplied to the remaining data. This way users can gain better context-sensitive insights. Bigfoot [290] uses an internal classification algorithm for learning common Border Gateway Protocol (BGP) routes over the Internet. They achieve this by constructing a shape, consisting of the route and a line from its end to its begin, and comparing it to polygons classified as normal.

Internal models enable users to adjust the definition of an alert and reapply it to the data set. Wagner and Healey et al. [139, 326] mention the need for security visualization to become more flexible and to better intertwine analytical methods with visualization. In recent research, more and more visual analytics solutions [12, 103, 196, 341] have been proposed.

2.9.4. Taxonomy Overview

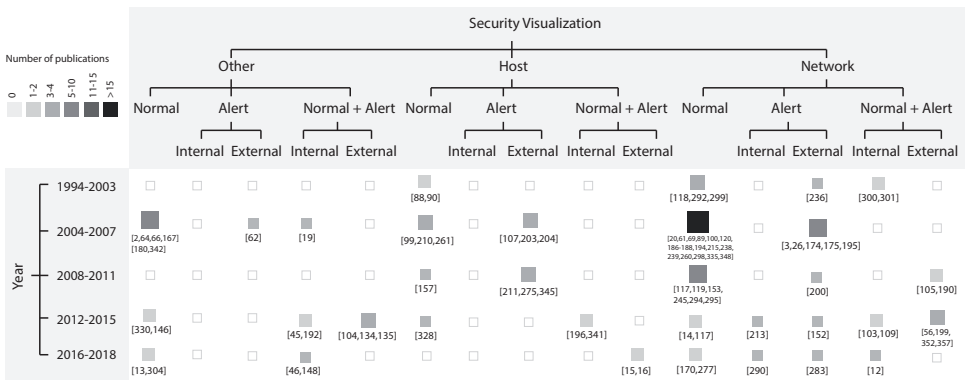


Figure 2.12: Security visualization taxonomy. The rectangles show the number of publications per category.

Figure 2.12 shows a graphical overview of the proposed taxonomy. The rectangles show the number of publications per category. Most of the security visualization systems have been designed in the period 2004-2007. They were typically focused towards the design of a new visualization by proposing a unique scheme to visually encode normal network events or external alerts in a visualization of choice. After the survey of Shiravi [274] in 2012 more visual analytics systems have been proposed to gain insight in internal alert collections. In addition, visualizations are no longer focused towards the visualization of either alerts or normal traffic. Systems such as OCEANS [56], Lamagna et al. [190], and Ocelot [16] combine PCAP traffic, IDS logs, and Netflow records to obtain a more complete overview of the network. Still, the analysis of PCAP traffic is currently limited to either flow analysis or the exploration of specific properties of protocols such as DNS [261] or BGP [300]. In this dissertation we explore DPI traffic in greater detail.

2.9.5. Taxonomy Event visualization

Event visualization is not limited to security applications, but is also a popular topic in domains such as healthcare [263] and business analytics [316]. Rind et al. [263] for instance



Figure 2.13: a) Systems such as Keshif [343] focus on the discovery of patterns in multivariate data records whereas b) Coquito enables the analysis of multivariate data in event sequences.

provide an overview of event visualizations that have been used to analyze Electronic Health Records [142]. A survey by Du et al. [82] describes different strategies that analysts can use to reduce the data volume and pattern variety in event sequences. Wongsuphasawat et al. [336] elaborated on a number of data operations that are considered useful for the analysis of temporal event sequences.

Figure 2.14 shows an overview of data operations that are supported in existing systems and are considered relevant for the exploration of event sequences. More complex operations such as data alignment or traditional data clustering can be obtained using a combination of these operations. This is also illustrated in Figure 2.15.

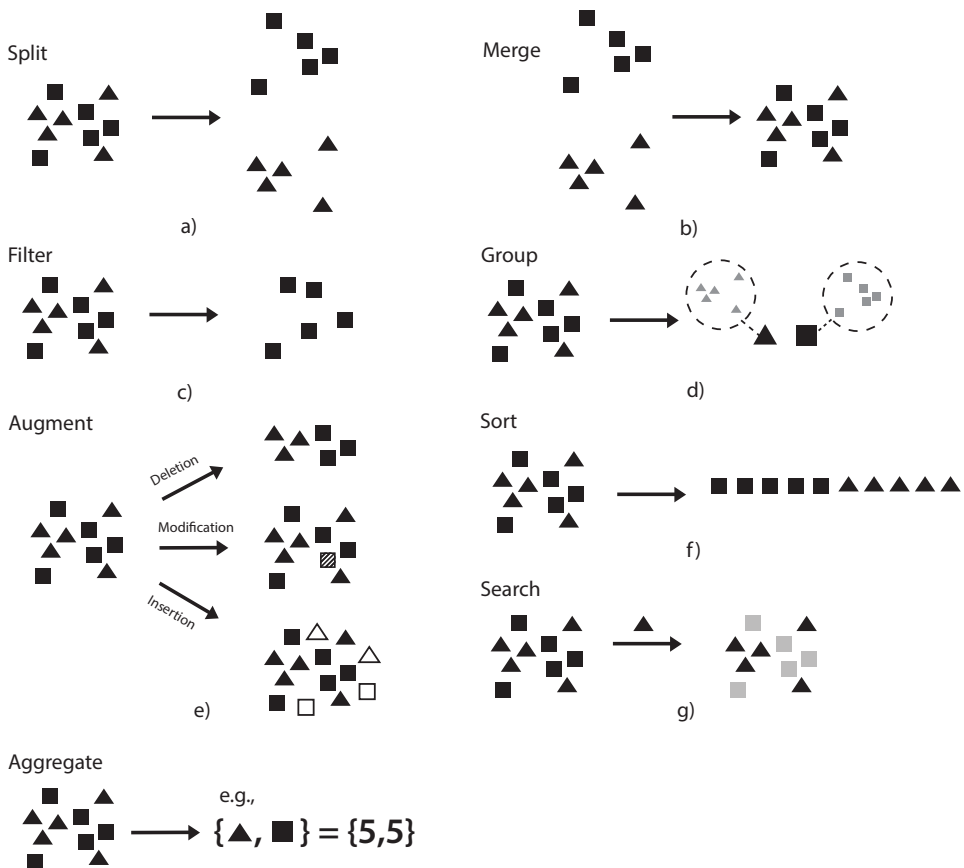


Figure 2.14: Overview of data operations that can be applied to event sequences or event data in general. Apart from data aggregation, all operations can be used as intermediate steps to perform more complex operations, such as data alignment or distance-based clustering.

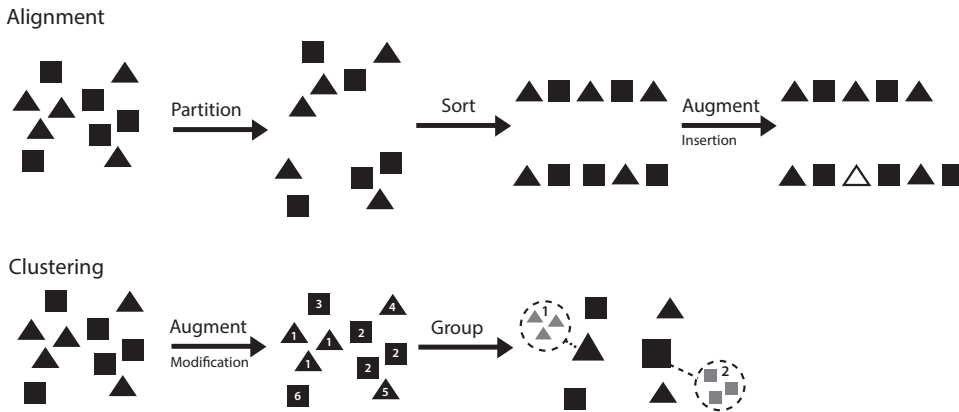


Figure 2.15: Operations such as data alignment and clustering can be achieved by combining operators such as augmentation, partitioning, sorting, and grouping.

2.9.6. Sequences vs. Records

Event visualizations typically focus either on the visualization of event *sequences* or the visualization of individual event *records*. The aim of event sequence visualizations is to discover patterns between event occurrences. This includes the analysis of sequential behavior (e.g., event ordering) and their (relative) time of occurrence. LifeFlow [338] for instance generates an overview of electronic patient health record sequences by ignoring time information between events. The result is an icicle plot showing where overlap between patient treatments exists using filtering and alignment. Other systems such as MatrixWave [356] visualize similarities and differences between sequences by showing this overlap using a matrix visualization (Figure 2.16).

Event record visualizations focus on the discovery of commonalities and differences between individual events. Events of interest can for instance originate from the same user, are sent at the same moment in time, generate the same error code, etc. Especially when dealing with multivariate data, the task of discovering relevant patterns in this wealth of information becomes a nontrivial task.

Tableau [230] and Keshif [343] are examples of systems that focus on the analysis of multivariate data records in general (Figure 2.13a). Systems such as TimeSlice [355] enable comparison between event records spread over time by enabling users to define a hierarchy of queries over the dataset, which are used to select events to display on the corresponding timelines. Crystalball [58] tries to predict future Twitter events by visualizing current events according to their time of occurrence, location, and topic.



Figure 2.16: MatrixWave displays event sequence collections by visualizing the overlap in each position in a matrix visualization. Matrices are positioned in a zig-zag pattern to represent the sequences from left to right.

2.9.7. Univariate vs. Multivariate

Visualizations that primarily focus on the visualization of event types, timestamps, and topological information are referred to as *univariate*, since they do not take additional metadata into account throughout exploration.

Eventflow [223] (Figure 2.17) for instance enables users to transform an entire dataset of temporal event records into an aggregated display, allowing researchers to analyze population-level patterns and trends in event types. Also Coco [208] limits the exploration of event data to event types and outcomes only. In order to cope with a wide variety of event types, Unger et al. [312] use automated pattern mining techniques to rank transitions in event sequences representing categorical states in lake sediment cores.

Other systems do take the metadata of events into account. These are also referred to as multivariate event visualizations. Liu et al. [202] uses automated pattern mining techniques to extract the most common user action sequences in click stream data. Beside type and timestamp, ClickstreamVis also enables users to discover patterns in device type, error codes, and browser information. Systems such as (s|qu)eries [351] and Coquito [181] enable users to search for multivariate sequential patterns in cohorts through visual querying (Figure 2.13b). (S|qu)eries supports querying events via regular expressions on multivariate data associated with events, whereas Coquito can incrementally construct cohort selections by dragging and dropping constraints in a node-link diagram.

2.9.8. Low vs. High-dimensional

The categorization in Section 2.9.7 classifies event visualizations whether they focus on a single attribute or multiple attributes. The dimensionality of a visualization refers to the extent to which these visualizations explore the full attribute space.

The literature defines high-dimensionality of event data in two ways. Traditionally, event data is considered high-dimensional if they (besides type and timestamp) consist of many attributes. However, Gotz et al. [122] also define event data as high-dimensional if they contain a large variety of event types. Especially in Healthcare records, event types often represent encodings of multiple attributes such as diagnoses, treatments, medications, and lab tests. As a result, the number of event types can be in the order of thousands and more, making exploration a difficult task.

If there are a few attributes, glyphs can be used. The traditional piano roll glyph display [85] and PlanningLines [5] for instance visualize events as glyphs on timelines. Event types are encoded in the color and shape of the glyph. This method however does not scale well when dealing with thousands of event types or attributes.

DecisionFlow and Synopsis [57] are examples of visualizations that can process large amounts of event types in a single attribute. Decisionflow enables the analysis of electronic health records with over 3,500 event types by combining an incremental milestone-based data representation with statistical analysis and an interactive flow diagram. Frequency [246] and Synopsis [57] try to visually summarize high-dimensional sequences using automated pattern detection techniques. Synopsis uses a two-part representation. The first part automatically discovers overlap between the sequences. The second part visualizes corrections (e.g., event insertions and deletions) in a separate view to enable full reconstruction of the original sequence. In Chapter 5 we present a visualization technique to visualize large amounts of attributes in a multivariate event visualization.



Figure 2.17: a) EventFlow [223] is an extensive novel tool to summarize univariate event sequences, search for temporal patterns, and apply data transformations to gain new insight in event collections. b) Decision-flow [122] enables high-dimensional analysis by introducing a milestone-based aggregate data structure and corresponding temporal query methods.

2.9.9. Taxonomy Overview

2

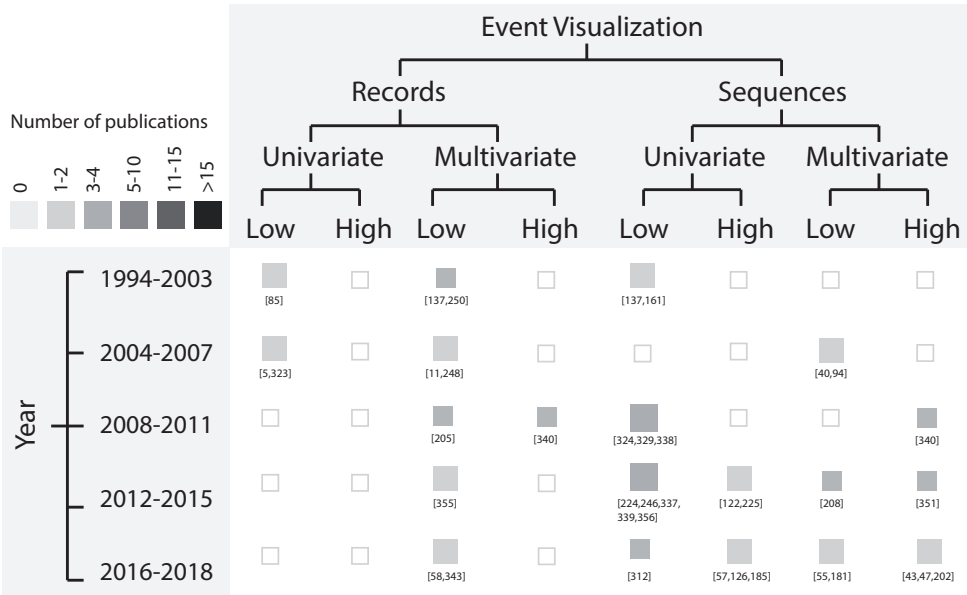


Figure 2.18: Event visualization taxonomy. The rectangles shows the number of publications per category.

Analogously to the taxonomy for security visualizations, Figure 2.18 shows a graphical overview of the taxonomy for event visualizations. Most of the event visualization techniques are focused towards the analysis of temporal patterns in event sequences. Especially starting from 2008 many event sequence visualizations focused on the analysis of event types and timestamps. Monroe et al. [225], Wongsuphasawat et al. [339, 340], and Wang et al. [329] proposed different techniques to effectively search for patterns in univariate event sequences.

Univariate record visualizations are typically limited to the visualization of a single attribute by plotting the events in a scatterplot or on a timeline. The piano roll displays of Vrotsou et al. [323], Eick [85], and Aigner [5] mainly focus on discovering where and when particular events have happened rather than studying commonalities and differences between sequences of events.

In the last few years, more focus has been on event visualizations to discover patterns in high-dimensional and multivariate data. Similar to security visualizations, Wongsuphasawat et al. [340] use ranking and similarity metrics to find comparable categorical records and similar event sequences. To this end, these systems have been categorized as both record and sequence visualizations.

Boundaries between categorizations are not strict. Especially the distinction between high-dimensional and low-dimensional visualization can be difficult. Most of the papers that claim to analyze high-dimensional data were either working with 1,000 event types or more or analyzing more than 10 different attributes besides type and timestamp. We therefore

aimed to classify the papers according to these guidelines.

2.10. Bridging gaps between Event and Security visualization

2

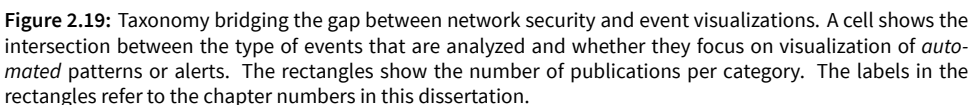
The variety in security visualization surveys and terminology makes it difficult to obtain an overview. From a data perspective however, many of these data sources can be simply modeled as event collections. To bridge the gap between security and event visualization, we propose a comparison of the techniques by combining the taxonomies in Section 2.9.4 and 2.9.9. Figure 2.19 shows an overview of all analyzed event and security visualizations. References to work in event visualizations are underlined, references to security visualization are not. The contributions of this dissertation are highlighted in red in Figure 2.19. The numbers in the blocks refer to the chapter numbers where each contribution is covered.

Most of the event visualizations focus on the visualization of univariate data. In contrast to security visualizations, few event visualizations focus on the analysis of individual records. Similarly, few security visualizations focus on the analysis of event sequences.

Event sequence visualizations in general are classified as host-based events, since most of the visualizations assume that events have a source or `sequence_id`. In addition, research by Unger et al. [312], Wongsuphasawat et al. [340], and Kwon et al. [185] were classified as alert based visualizations, since they use automated techniques to determine the relevance of event sequences and rank or cluster them accordingly. Security systems by Huyn et al. [148], Angelini et al. [12], and Liao et al. [200] also use this kind of functionality to determine the severity of an alert or to gain insight in larger alert collections.

The taxonomy illustrates that the amount of overlap between security and event visualization is surprisingly small. Most of the rectangles either exclusively consist of references to security or event visualizations. However, security visualizations in the categories `host` and `other` can also be used to study event data. Especially for the visualization of multivariate data in event records, the systems by Hao et al. [134], Keim et al. [167], and Walton et al. [328] can be useful. Vice versa, systems such as (s|qu)eries [351] or ClickstreamVis [202] in turn can be very useful in studying sequential patterns in security event logs.

The taxonomy in Figure 2.19 shows that existing security and event visualizations cover a broad spectrum of the proposed categorization. However, for CERTs and SOC members there are still challenges in the area of visualization and in particular visual analytics. Most of the security record visualizations do not support the analysis of Deep Packet Inspection data or do not use internal automatic network intrusion detection algorithms to assist in the detection of anomalies. Furthermore, there are no visualizations that focus on the exploration of high-dimensional data in sequences of network traffic. Although (s|qu)eries [351] enables users to visually query data for patterns, they provide little support for the exploration of unknown patterns or attributes of interest using visualizations or automated techniques. In the next chapters we investigate how to gain better insight in event collections by enabling internal automated support and sequential analysis in high-dimensional multivariate network traffic.





Monitoring Multivariate Event Collections

3

This Chapter is based on [45]:

B.C.M. Cappers and J.J. van Wijk. SNAPS: Semantic Network Traffic Analysis through Projection and Selection.
In Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec) 2016, pp. 1-8

3.1. Monitoring Multivariate Event Collections

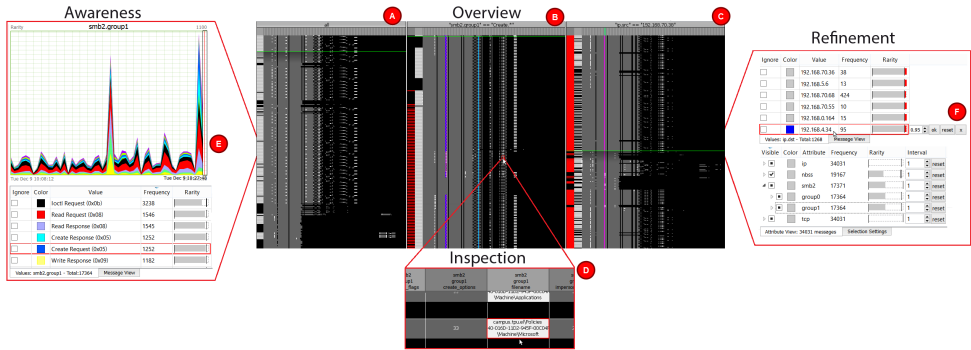


Figure 3.1: Network traffic exploration at the level of semantics through the creation of three selections of interest in parallel.

Most network traffic analysis applications are designed to discover malicious activity by only relying on high-level flow-based message properties. However, to detect security breaches that are specifically designed to target one network (e.g., Advanced Persistent Threats), Deep Packet Inspection and anomaly detection are indispensable. In this chapter, we focus on how we can support experts in discovering whether anomalies at message level imply a security risk at network level. In SNAPS (Semantic Network traffic Analysis through Projection and Selection), we provide a bottom-up pixel-oriented approach for network traffic analysis where the expert starts with low-level anomalies and iteratively gains insight in higher level events through the creation of multiple selections of interest in parallel. The tight integration between visualization and machine learning enables the expert to iteratively refine anomaly scores, making the approach suitable for both post-traffic analysis and online monitoring tasks. To illustrate the effectiveness of this approach, we present example explorations on two real-world data sets for the detection and understanding of potential Advanced Persistent Threats in progress.

3.2. Semantic Network traffic Analysis through Projection and Selection

One of the main challenges in the area of network traffic analysis is how to detect when a network is being exploited. Especially for critical infrastructures, such as power plants [60], hackers nowadays are willing to design complex viruses to maximize the damage in one specific infrastructure. The main difficulty with Advanced Persistent Threats (APTs) [280] is the involvement of domain knowledge such that their traffic can no longer be distinguished from regular activity by simple inspection of high-level properties, such as message length and destination address. Current methods [66, 174, 175, 215] focus on the analysis of these properties, because in practice they have shown to be sufficient for the discovery of traditional attacks [71, 214]. The fact that these techniques consider traffic content as a black box makes them unaware of anomalies at the level of semantics.

To ensure that systems and data in the network are secure from APTs, the content of the traffic has to be taken into account. For example, we are not only interested in which host sends packets to a particular host, but we are also interested in whether the action inferred by these messages represents the access to an uncommon function call or file in the network. The heterogeneity and abstraction level of the data makes it very difficult to decide if a message is truly malicious. We believe that the greatest insights can be obtained by comparing anomalies to similar parts of traffic and try to understand how they differ from each other with respect to context and structure. In order to gain this insight, we propose a new approach that enables security experts to discover high-level security risks, starting from a collection of automatically classified low-level anomalies, through the use of selection and projection. More specifically, our main contributions are:

- A novel exploration method for the analysis of raw network traffic, enabling the expert to inspect and compare specific parts of the traffic in parallel while preserving context;
- A tight coupling of machine learning and visualization that assists experts in detecting malicious traffic, through iterative refinement of classifier parameters;
- The ability to gain statistical insight in how messages differ from regular traffic and why a message was classified as malicious.

The chapter is structured as follows. First, related work is discussed in Section 3.3. Next, the scope and approach for traffic analysis is discussed in Section 3.4. In Sections 3.5, 3.6, and 3.7 an overview of the system is presented after which visualization, classification, and interaction are described. In Sections 3.8 and 3.9 we provide two example explorations on real-world data sets and discuss the limitations of the approach. Conclusions and future work are presented in Section 3.10.

3.3. Related Work

Network traffic analysis is an extensively studied topic, covering a wide range of techniques. We give a broad overview first, followed by a detailed discussion on pixel-based visualization techniques.

3.3.1. Data

From a data perspective, current analysis techniques can be grouped into two categories: *byte-oriented* and *attribute-oriented* analysis.

In byte-oriented analysis, network messages are considered as a sequence of bytes enabling visualization techniques to analyze the full payload of a network message. These visualizations typically provide insight in the traffic by encoding the byte sequences in text or pixels. The binary rainfall [66], digraphs [65] and malware images [231] are well-known examples in this category. Anomaly detection systems for this type of analysis typically rely on byte distributions and pattern matching to discover undesired content. Since byte sequences do

not contain any information about which bytes together represent an attribute in a message, these detection methods often work poorly for anomalies at the level of attributes.

In attribute-oriented analysis, messages are dissected according to their *protocol* structure, thereby gaining knowledge about the actual values that were sent in the message. The result of dissecting a message typically is a collection of attributes and values. The presence of an attribute or value is determined by the type of network message, thereby significantly increasing the heterogeneity of the data. Current methods often limit their analysis to high-level protocols such as TCP and IP, thereby only relying on common flow-based attributes such as IP addresses, port numbers, and message lengths [174, 210, 352]. For a more complete overview, we refer to Chapter 2.

There are also examples where both byte structure and attribute analysis are taken into account. For instance, the open source application Wireshark [63] is an extensive protocol analyzer that can dissect network packets and display the payload in a (hierarchical) textual representation. Especially for debugging applications, the wealth of information provided by Wireshark can help the expert to analyze traffic in great detail. The software unfortunately does not assist the expert in finding anomalies and can become a burden when analyzing or monitoring large network samples.

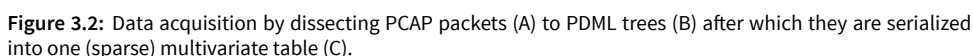
3.3.2. Visualization

In SNAPS we use a pixel-based visualization that conveys the global structure of network messages as well as anomalies in that structure. A message is displayed as a horizontal sequence of pixels. Pixel-based visualizations have been used often for network traffic analysis, some examples are:

- Binary rainfall [66] by Conti et al. visualizes network messages as a single line of pixels where pixels are colored based on protocol type, various byte encodings, and frequency. They showed that the visual encoding of network traffic does not have to be complex in order to discover nontrivial patterns. Their byte-oriented approach unfortunately makes the method unsuitable for the detection of APTs.
- PortVis [215] by McPherson et al. uses a color-based grid visualization to visualize the amount of network activity between port numbers. By using a zoom lens, the user can obtain port number information to trace back the cause of the anomaly.
- IDS rainstorm [3] by Abdullah et al. visualizes Stealthwatch [191] intrusion detection alerts by showing the severity of alerts over time using a set of rectangular regions that represent a large continuous range of IP addresses.

Previous methods construct an image to represent the values for one or two attributes in the data. To cover the wide variety of attributes, in SNAPS we construct an image to represent the full range of attributes.

A method specifically designed for multivariate data exploration and closest to our technique is the Pixel Carpet visualization by Landstorfer et al. [192]. In this visualization every log record is visualized as a stack of pixels, where every pixel denotes the frequency of a



In summary, current methodologies are either focused on the visualization of high-level message attributes or the visualization of unstructured low-level representations. Current methods that do consider message attributes typically consider only a few flow-based attributes.

With the vast amount of information that is sent over networks, one of the main concerns is to know when something undesired is being sent. Especially for critical infrastructures, the presence of malicious traffic can have severe if not life-threatening consequences.

The involvement of domain knowledge in APTs makes the infiltration of these viruses (typically through social engineering) in networks nearly impossible to prevent. Once the threat is established, we can analyze unencrypted internal network traffic for anomalies that arise during the APTs exploitation phase. As a consequence, traffic from or to external sources is considered outside this scope.

In practice, it is possible for messages to consist of an arbitrary number of protocols, where one protocol can even occur multiple times. Since semantic attacks happen at the application layer of the network protocol, we restrict the analysis of messages to the following protocols: (DCE)RPC, SMB2, and S7 [278]. (DCE)RPC are application protocols to send remote proce-

dures calls over a network. SMB2 is typically used for file management in a network, whereas S7 is a classified industrial control protocol by Siemens for controlling low-level hardware components. We use ETH, IP, and TCP protocol information to trace the anomalies back to physical entities in the network. To avoid the significant increase in attribute space due to protocols occurring multiple times, analysis of messages is limited to the first occurrence of every protocol.

3

3.4.1. Data acquisition

Before we can analyze network traffic in greater detail, we first need to analyze network traffic with protocol semantics. We use the Wireshark dissector to convert a raw network message to a so-called Packed Detail Markup Language (PDML) parse tree, describing on a per protocol basis the values and attributes that are present in that message. Figure 3.2b shows an illustration of how PDML trees are structured. Attributes in a protocol are structured hierarchically. In general, network messages consist of multiple protocols, each with their own purpose and different level of abstraction. Depending on the protocol semantics, attributes in protocols can represent numerical ranges (e.g., `tcp.srcport`), strings (e.g., `ip.src`), or boolean values (e.g., `tcp.flag.SYN`). The presence of a protocol, attribute, or value not only depends on the type of message, but also depends on the context in which the message was sent.

More formally, let S represent the set of attributes that the expert wants to analyze. Furthermore, let $PDML(m)$ represent the PDML tree of message m . Attribute $A \subseteq PDML(m)$ if and only if there exists a path $P = [p_0, p_1, \dots, p_n]$ from root to leaf in $PDML(m)$ such that $p_0.p_1 \dots p_n = A$. A is also referred to as the *serialization* of P . Finally, let $val(A)$ represent the value stored in p_n of A . We can now model a message m as a set of (A, v) pairs such that:

$$A \in S, \text{ and} \quad (3.1)$$

$$v = \begin{cases} val(A) & \text{if } A \subseteq PDML(m) \\ \text{undefined} & \text{otherwise} \end{cases} \quad (3.2)$$

Since the set of all possible attributes is too large to analyze, in SNAPS we use domain knowledge and a large sample of network traffic to determine which attributes are worth analyzing. The resulting table after serializing the collected PDML trees can be found in Figure 3.2c. Since the set of possible attributes is much larger (500 or more) than the set of attributes contained in a message (order of 10s), the data can become quite sparse, increasing the complexity of visualizing the payload data.

Using our previous model, we can now formulate the analysis task for the detection of APTs as trying to gain insight in the presence, description, temporal behavior, and rarity of these (attribute, value) pairs. The serialization of PDML trees to one multivariate table enables us to compare differences and similarities between messages at the level of attributes. Since the presence of one attribute depends on the presence of other attributes, showing values of multiple attributes simultaneously enables us to gain insight in these dependencies. In order

to make the visualization of alerts and patterns feasible for a large number of attributes, we chose a pixel-based visualization approach.

3.5. SNAPS: Selection and Projection

Network traffic exploration is a challenge due to the large amount of data that is being generated in a relatively short period of time. Furthermore, the heterogeneity and complex structure of the traffic content adds a new dimension to the analysis of network traffic. We cannot expect the expert to know the meaning of every dissected value or attribute. However, the expert should be able to determine the severity or cause of an anomaly by *inspecting* and *comparing* similar type of messages in different contexts. To support this, we need a scalable interactive method to simultaneously explore low-level anomalies, while maintaining a high-level *overview*.

We tackle the scalability problem by visualizing network traffic using a pixel map [164]. The high “data-to-space” ratio of pixel maps enables us to visualize large amounts of attributes and network messages in a limited amount of screen space. Furthermore, to maximize the speed of analyzing traffic, we aim for a computationally cheap classification method using histograms. The level of granularity in which traffic is analyzed is determined by filtering on attributes or values in the traffic. In the world of relational algebra [32], these operations are referred to as *projection* and *selection* respectively. To enable the simultaneous exploration of traffic in local and global contexts, we do not limit the exploration to one selection, but to a number of selections of interest (see Section 3.5.1) enabling the expert to:

- *Drill down*: inspecting alerts against different subparts of the network, while remaining aware of the rest of the traffic, or
- *Scatter*: creating multiple views to keep an eye on critical or suspicious entities in the network (e.g., hosts, files).

To tackle the problem of dealing with large amounts of false positive alerts, we use a human-in-the-loop approach [264] that enables the expert to inspect and *refine* classification results on a per selection basis. By means of *color rules*, the expert is able to highlight specific events in the traffic for which the severity is already known. Figure 3.3 shows a schematic overview of the SNAPS exploration process. When trying to find potential virus attacks, time is of the essence. The earlier anomalies in the network can be detected, the faster we are able to manage the attack. For this reason, we designed the system in such a way that it is suitable for both post-traffic analysis and live monitoring. Although the traffic dissection by Wireshark is rather computationally intensive to be used for real-time monitoring, there are (more complex) alternatives, like the Bro dissector [244], that are suitable for obtaining near-real-time dissections. To assist the expert in exploring and explaining traffic alerts, we use five coordinated views as depicted in Figure 3.5. For each view we describe its functionality and design decisions. For the demonstration of the functionality in practice, we refer to the supplementary video ¹.

¹<https://www.youtube.com/watch?v=aYywTOYjYDA>

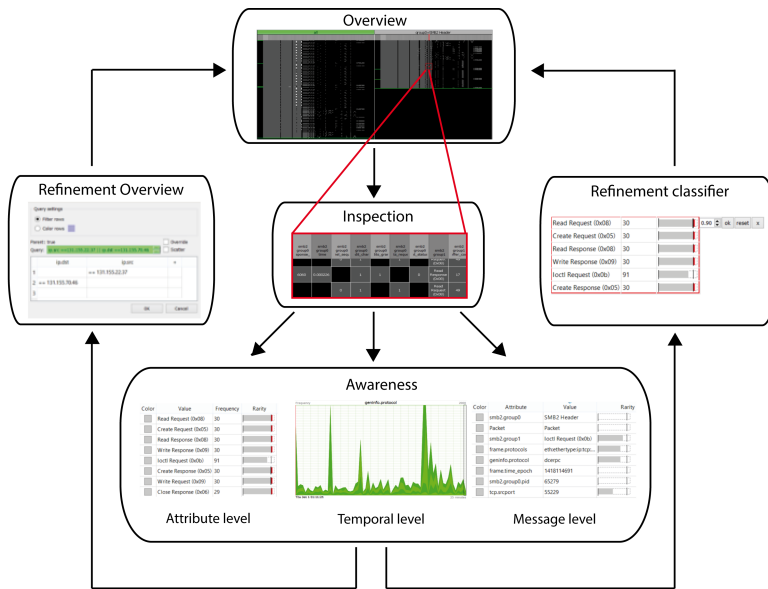


Figure 3.3: The SNAPS workflow model for network traffic exploration. The expert uses the overview to monitor the presence of alerts in the selections of interest. Upon the discovery of an alert, the expert tries to gain more insight by inspecting its location and value. Depending on the familiarity and rarity of the inspected value(s) and attribute(s), the expert assesses the severity of the alert by iteratively inspecting: the occurrence of other values in the message; the value distribution of an attribute; or the presence of a message value over time. Depending on his findings, the expert either decides to ignore the alert, prevent the alert from happening by refining the classifier, or refines his selections of interest to analyze the alert in a different context.

3.5.1. Pixel viewer

For every selection of interest, the pixel viewer visualizes message payload by creating an image where the horizontal axis represents the attribute space the expert is interested in and the vertical axis represents the collection of network messages. The result is that every message corresponds to a single line of pixels, where the brightness of a pixel p_{ij} represents the rarity of message i at attribute j according to Section 3.6. Since it is hard to distinguish colors for small objects [289], we use a discrete grayscale map (Figure 3.5) consisting of three colors: pixels are colored black if the message does not contain the corresponding attribute, gray if the value in that message is not considered rare and white if the value in the message is considered to be rare. The rarity of a message as a whole is visualized by prepending the image with an additional column. Values and attributes in messages become visible by inspecting pixels with a zoom lens (Figure 3.5b).

Besides the grey shades that indicate rarity, a subtle hue can be added to message attributes to indicate different protocols (Figure 3.4). Besides coloring attributes, we enable the expert to discover patterns by coloring pixels according to their value or more complex expressions. To improve the distinction between pixels and prevent pixel colors from spreading to their neighboring cells, tiles of 2 by 2 physical pixels are used instead. As soon as an incoming

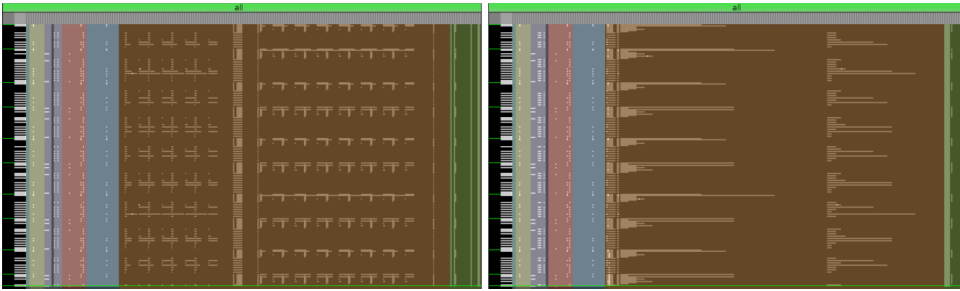


Figure 3.4: Attribute ordering without (left) and with (right) frequency clustering. Attributes are colored according to their protocol.

message adheres to some color rule r , SNAPS creates a marker in front of the pixel view whose color corresponds to r . In situations where multiple color rules apply, SNAPS creates a marker according to the first matching color rule. Figure 3.5 shows an example how coloring is applied.

Similar to the pixel visualization by Conti et al. [66], we use the notion of a radar to replace old messages with new ones. In contrast to traditional scrolling, previous messages are not shifted at the arrival of new data, thereby making the visualization more stable when analyzing traffic at a larger pace. A direct consequence of visualizing messages in a sequential fashion is that the inter-arrival time between messages is no longer visible. To make the expert aware of these changes in flow, the radar creates a green marker in front of the image whenever the timestamp difference between messages is larger than a second. The distance between green markers is an indicator for the amount of traffic that is being sent in between timestamps. For more detailed information about temporal behavior, the expert can use the Time view (Section 3.5.2).

Attribute ordering

When combining multiple PDML trees into one attribute space, the ordering in which attributes should be positioned is not uniquely defined. To illustrate the latter, consider two message m_1 and m_2 with attribute sequences $[A, B, C]$ and $[A, B, D]$ respectively. Although the ordering of attributes in a message depends on its structure, when combining the attribute space of two messages into one, it is undefined whether attribute C should precede D or vice versa. Although this ordering does not influence the classification result of a message, it can help the expert to localize attributes in the visualization more quickly. One way to solve this is to sort attributes alphabetically. Since the hierarchy is implicitly stored in the attributes, sorting attributes alphabetically causes the attributes with the same PDML paths to be grouped together. Any logical ordering between siblings (e.g., header attributes before payload), unfortunately, may no longer be preserved. To solve the second issue, we sort the siblings within each group according to their frequency. The effect of sorting attributes with and without frequency analysis is illustrated in Figure 3.4.

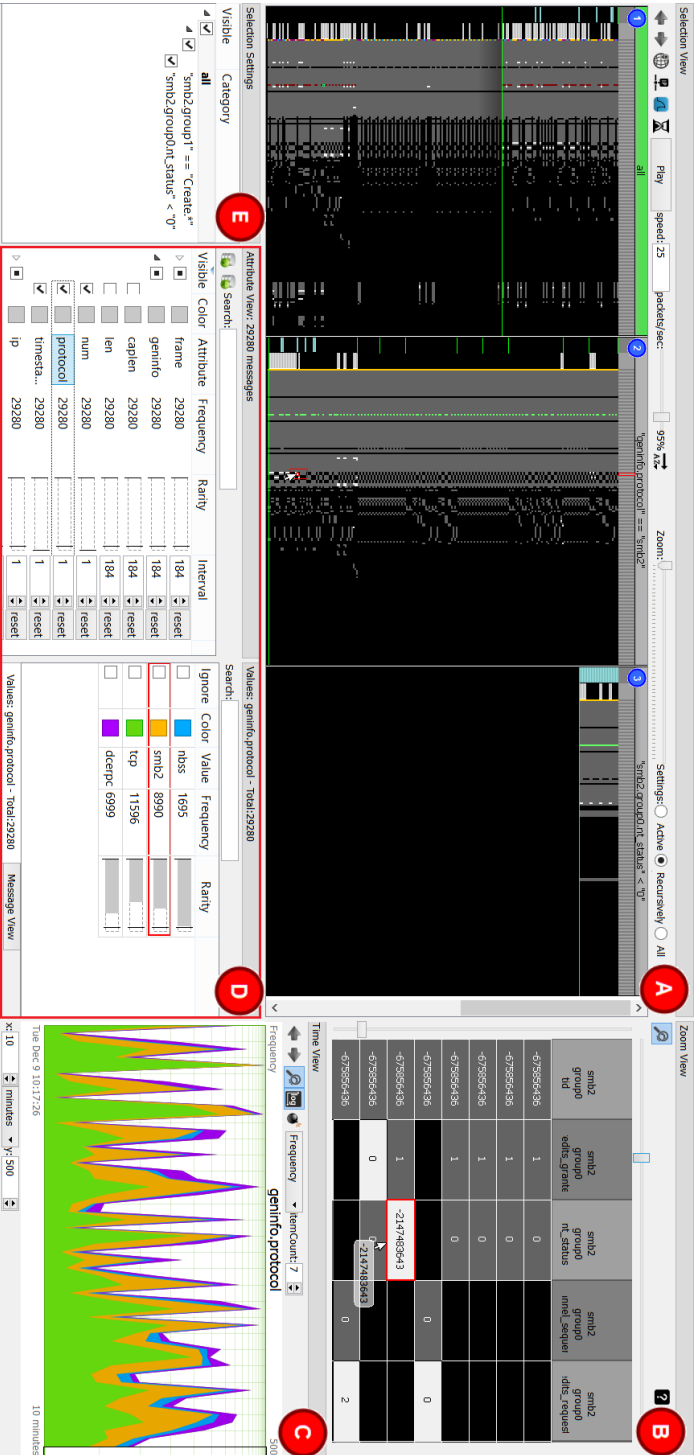


Figure 3.5: Graphical user interface of the implemented system: A) Pixel view showing parts of the network traffic within the scope of selections and projections of interest. Settings with respect to the scan speed and coloring and ordering of attributes are adjusted using the controls in A. B) Lens view for the inspection of message values and alerts in the pixel view. C) Depending on the selected pixel view and attribute, the Time view shows the attribute distribution over some specified time range. Settings with respect to axis scaling, coloring of values, and time range are adjusted using controls in C. D) Attribute view enabling the expert to inspect value distributions of attributes and refine the classifier by modifying rarity thresholds or removing predominating values from the histograms. E) Selection view showing an overview of current selections.

Selections and Projections

As mentioned earlier, selection and projection enable the expert to inspect anomalies against different parts of traffic while focusing on a specific subset of attributes. A downside of applying filters is that the expert is no longer aware of alerts that were present in previous settings. One can imagine, however, that an expert wants to keep an eye on specific entities in the network (e.g., a critical host) while staying aware of other activities. To prevent the expert from losing context, we enable the expert to create multiple selections of interest by creating multiple pixel views in parallel.

When creating a new view B , by default, projection, selection, and color settings are inherited from view A where the filtering was initiated. This enables the expert to continue the exploration without having to reapply every setting in the new view. The histograms for view B are constructed by revisiting the network traffic within data window ω , only considering messages that are valid with respect to the current selection. Network messages in B are visualized in the pixel view if and only if these messages are also visually present in A . This way, messages in B initially are always a subset of the messages in A enabling the expert to see the impact of applying a new selection of interest. By means of the Time view, the expert can revisit earlier parts of traffic that are outside the scope of the pixel view.

In order to gain insight in anomalies within a specific selection of interest, the expert is enabled to train a separate classifier for that selection. Since highly specific selections may result in inaccurate alerts due to overfitting, by default, alerts with respect to the histograms of the parent view are shown in the visualization. Since both anomaly scores are maintained in parallel, the expert can toggle between *local* and *global* anomaly scores. An overview of the current selections is shown by means of a tree structure (see Figure 3.5e). The expert can add, remove, show or hide selections whenever necessary. We enable the expert to apply new settings simultaneously to all views, the selected view or the selected view along with its descendants.

3.5.2. Time view

The Time view shows an overview of the number of messages that are sent over the last n time units. By selecting an attribute A , the expert is enabled to inspect the distribution of the values of A over various periods in time. Depending on the selected pixel view, only messages that are valid with respect to that selection are shown in the Time view. Upon the arrival of new data, the line chart is shifted to the left, causing messages older than n time units to be no longer visible. To prevent the chart from cluttering, only the top m values in A are shown that are either most frequent, most rare, or selected by the expert. Remaining values are grouped in a miscellaneous category.

The Time view enables the expert to scroll back to earlier parts of traffic. To make the expert aware of the time interval that is spanned between the oldest and the newest message in the selected pixel view, a black window is rendered in the Time view. Since the time interval of the pixel view depends on the inter-arrival time between messages in that view, the width of the visualization window may vary over time. The expert can scroll back to earlier parts

of traffic by dragging the visualization window along the time axis after which the selected view and all descendants are updated. Here traditional scrolling is preferred over a radar, since experts can determine their own rate in which the views have to be updated.

When scanning messages sequentially, there is a choice between *message-oriented* versus *time-oriented* scanning. In message-oriented scanning, messages are scanned at a fixed rate causing the pixel visualization to be updated at a constant rate. Since the inter-arrival time between messages is not taken into account, the update rate of the Time view varies over time. In time-oriented scanning, messages are grouped in fixed time intervals such that the Time view is updated at a regular pace. Since multiple messages can adhere to the same time interval, the refresh rate of the pixel views is no longer constant. In case of for instance traffic bursts, message-oriented scanning may be preferred over time-oriented scanning if the data contains samples of malicious activity (e.g., file-scan). If a data burst is not of interest, the expert can switch to time-oriented scanning to analyze these messages at a higher rate.

3

3.5.3. Attribute view

The Attribute view (Figure 3.1d) enables the expert to inspect the frequency and rarity of attributes and values that are present within data window ω . Depending on the selected pixel view, only messages that are valid with respect to the pixel view selection settings are visible in the Attribute view. The tree structure on the left shows an overview of all attributes in ω by taking the union of all PDML paths in the PDML trees of messages in ω . The expert can adjust the projection settings of the pixel view by changing the visibility of attributes using checkboxes. Only attributes that occur in the projection settings are taken into account during classification. When selecting an attribute, the table on the right shows the value distribution of that attribute. The expert can inspect the frequency of values by means of sorting and filtering. Besides frequency, the rarity of a value (see Section 3.6) is visualized using a bar.

One major problem of anomaly detection is that there is no intrinsic difference between a malicious value and a new incoming value. One can imagine however that the creation of a new file in the network is not necessarily harmful. To keep the number of false alerts in such attributes minimal, the expert can adjust the rarity thresholds of the histograms on a per value basis (or bin basis for numeric attributes). Dominating values can be removed from the histogram by means of a checkbox in front of the value. The rarity threshold that is applicable to a certain value is shown as a vertical bar in the previously mentioned rarity bars. Thresholds can be modified by either dragging the threshold in the bar visualization or filling an exact value in a popup (Figure 3.1f). To prevent the expert from having to adjust every threshold manually, he is enabled to select multiple values at the same time or to specify a global rarity threshold at the level of an attribute or pixel view. In other words, if there is no threshold set for a value v in (A, v) , the threshold for A is used instead. Alerts for specific values and attributes can be ignored during their classification by setting the rarity threshold to 100%. The effect of modifying a threshold is immediately reflected in the brightness of the pixels. The expert can save and load thresholds on a per pixel view basis through import and export functionality.

3.6. Classification

Due to the large number of attributes per message, it is difficult for the expert to manually spot anomalous values in the traffic. To assist the expert in finding these anomalies, a simple but effective histogram-based classifier is used. Histograms in general are computationally cheap to maintain, easy to understand and can be applied to both numerical and categorical attributes. Their ability to be updated in an incremental fashion makes them both suitable for offline and online analysis. Anyhow, the SNAPS approach is independent of the chosen classifier, and developing better classifiers is a topic for future work.

3.6.1. Model

In Chapter 2 we identified three types of anomalies, namely point, contextual, and collective anomalies. Network messages are considered point anomalies whenever they are anomalous with respect to the entire data set (e.g., the invocation of a deprecated function call). Messages that are only anomalous in a specific context (e.g., the access of a restricted file by an unauthorized user) are contextual anomalies. Collective anomalies are collections of messages that together are anomalous with respect to the entire data set. Since automatic collective anomaly detection methods are rather error-prone for highly heterogeneous and time-dependent data, they are considered outside the scope of this work. Instead, we provide the expert a Time view where collective patterns can be visually inspected over different periods in time.

In our online classification approach, network traffic is considered to be relevant within time window ω . Upon the arrival of new data at current time t , messages older than $t - \omega$ are removed from the window and replaced by new ones. For every incoming message, the classifier determines the rarity of values in that message after which the histograms are updated. When training a classifier on a new subset of traffic, the minimum size of the training set \mathcal{T} with respect to that subset is determined using Yamane's sample size formula [154].

3.6.2. Anomalies

In order to decide whether a network message is a point anomaly, we first have to define when a value in the message is considered to be anomalous. Let $T = (A, v)$ be an attribute value pair in message m . Let $\#A$ denote the number of messages in the data set with attribute A other than **undefined** and let $\#v$ denote the number of messages with (A, v) . T is considered to be *rare* if and only if:

$$1 - \frac{\#v}{\#A} > \tau \quad (3.3)$$

where τ represents a rarity threshold defined by the expert. In case where $\#A$ is smaller than Yamane's sample size with respect to \mathcal{T} , every value is considered to be rare, since the number of samples in this attribute is too low to build an accurate histogram. **undefined** values are excluded from the histograms, since they predominate the distribution of sparse attributes. To minimize the number of false positive alerts, values for numeric attributes are binned. By

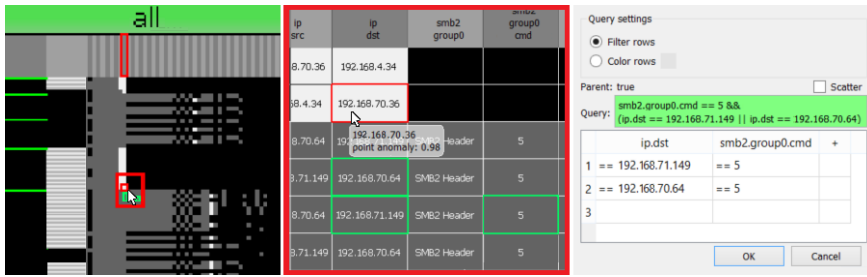


Figure 3.6: Multiselection of values and corresponding query.

default, the bin size s of a numeric attribute is computed by applying Scott's rule [271] on the training set after removing outliers:

$$s = \frac{3.5\sigma}{n^{1/3}}$$

Scott's rule is chosen for its simplicity, since we expect bin sizes to be refined during exploration.

For the detection of contextual anomalies, the expert is enabled to train a new classifier on a selection of interest (see Section 3.5.1). This enables the expert to inspect distributions and look for anomalies on a smaller subset of the traffic.

3.7. Interaction

To enable querying in SNAPS, three operations are supported:

- inspecting values;
- color messages according to rules;
- creation of selections of interest.

For the inspection of values, the notion of a lens is used, showing an enlarged part of the pixel visualization where additional information such as the values and attributes of pixels become visible. Upon the detection of an alert, the expert can stop the message scanning and lock the lens to inspect values in more detail. The rarity score of a value is shown by means of a popup (Figure 3.6). The expert is enabled to inspect the contents of a pixel in even more detail by switching to the Wireshark interface with one click of a button.

Visual coherence between views is achieved by using color. Hovering the mouse over a value highlights all messages in the pixel view with that value. Similarly, hovering the mouse over a message reveals the location of that message in other pixel views (Figure 3.1b).

Experts can create selections of interest using default, text-based and table-based filtering. To improve interaction speed, SNAPS provides default filtering functionality when the expert selects a pixel, value, or attribute. By means of context menus, the expert can choose to filter

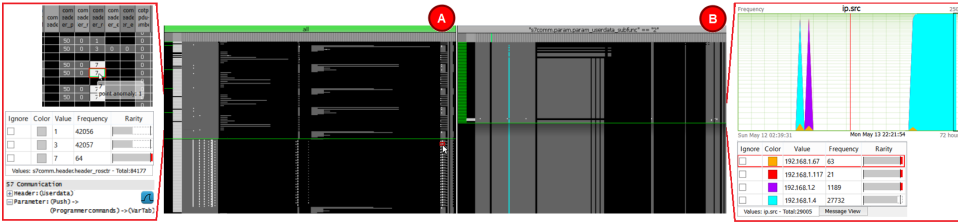


Figure 3.7: In view a) the sudden change in S7 traffic raises alerts in attribute `s7.rosctr`. The Wireshark interface shows that alerts are coming from commands to reprogram the PLC. Creating view b) only containing these commands shows that four machines were responsible for sending these commands in the last 72 hours.

the traffic by the presence or absence of the selected value, sending or receiving IP address, or by the conversation in which the message occurred. For more complex queries, a textual interface is provided to assist the expert in creating a query. When writing the query, the expert is instantly notified if the query is syntactically correct. Depending on the part of the query that is being constructed, the expert receives a list of possible attributes, operations, or values that were found in the selected view.

Since the usage of brackets in a query can affect the readability of a query in a negative way, an alternative form of visual querying similar to Excel's advanced criteria filtering [218] is introduced. Queries can be represented as a table where the columns represents attributes and a cell represents a condition (op, reg) where reg represents a regular expression and op the corresponding operation (either $=$, $!$, $<$, or $>$). Since a message can have at most one value per attribute, the query is constructed by taking the conjunction of all nonempty conditions in a row, after which a disjunction is taken over all rows in the table. Figure 3.7 shows an example of the resulting encoding. We can use the same encoding to enable the selection of multiple pixels into a new filtering condition. To prevent the selection border of two neighboring pixels from occluding each other, selection borders are drawn using a contour algorithm. For a more concrete overview of the interaction, we refer to the supplementary video in Section 3.5.

3.8. Use cases

To illustrate the effectiveness of our approach, we tested the application on two types of data sets. The first data set is obtained by recording one day of internal network traffic from a university. The data consists of approximately 400,000 messages and 500 attributes sent by 25 hosts, initially training the classifier on 30,000 messages corresponding to one hour of traffic. The second data set consists of approximately 500,000 messages, 650 attributes, 10 to 15 hosts representing one week of S7 traffic from a governmental industrial control system. For this data set, the classifier is initially trained on 100,000 messages corresponding to one day of traffic. In the use cases, ω is set to half a day and three days of network traffic respectively.

3.8.1. University

3

We initially start the exploration by scanning messages without any selections of interest. The network data contains a wide variety of TCP, DCERPC, and SMB2 traffic (view 1, Figure 3.5a). Since we are interested to find anomalies at application layer, we create a selection of interest *B* only containing SMB2 traffic (view 2, Figure 3.5a). To obtain more reliable classification results for SMB2 traffic, we switch to the local classifier trained on *B*. Finally, predominating values in the data such as `Ioctl` request are ignored during classification and the rarity thresholds of attributes like `smb2.file_name` were raised to reduce the number of false positives in the visualization.

After scanning 30 minutes of traffic data, in approximately 5 minutes, we noticed a group of anomalies in *B*. The lens view showed that the alert was raised by a non-zero value in attribute `smb2.nt_status` (Figure 3.5b). To receive automatic notification of this alert, a color rule for this condition is applied (indicated by the cyan markers). By creating a new selection of interest *C* for which `smb2.nt_status < 0` and specifying that the pixels showing the source IP address of such messages should be colored green, we can see that these, what turned out to be SMB2 buffer size warnings, were coming from the same IP address (view 3, Figure 3.5a). We close *C* and continue exploration.

Around 10:27 AM, an SMB2 burst was detected as depicted in Figure 3.1a. Selecting the `smb2.cmd` attribute in the Time view of *A* shows an increased number of files being created in the network (Figure 3.1e). By creating a selection of interest *D* only considering file creations, we obtain more frequent patterns (depicted in Figure 3.1b). During the burst, a group of alerts was spotted in the `smb2.file_name` attribute. Although it is common for that attribute to generate alerts (e.g., the creation of a file), the fact that these alerts were occurring quite fast after one another in *D* was suspicious. The IP address responsible for sending these messages was found by means of coloring. Creating a separate selection of interest for this address with respect to *D* and selecting the `ip.dst` attribute in the Attribute viewer, we obtain a list of all locations where these files have been created (Figure 3.1c). Hovering the mouse over address `192.168.4.34` highlights all locations of this value in the pixel views, showing that most files in the burst period were created at this location (Figure 3.1f). The values corresponding to the alerts represented a large collection of Microsoft group policy files being accessed in the network (Figure 3.1d). Since policy files store authorization access at network level, they can only be modified by network administrators. The interesting part, however, is that none of the users in the data set are administrators.

3.8.2. Industrial control system

In contrast to the office network data, S7 traffic in the governmental control system shows very regular patterns, suggesting that entities in the system send traffic within a particular ordering (Figure 3.7a). Based on the shape and values that arise from these “vertical histogram” patterns, we can see that the monitoring system `192.168.0.13` reads sensor values from components at a fixed pace. On May 13th 9:00 the pattern becomes disturbed, raising a large collection of alerts in attribute `s7comm.rsotv`. When switching to the Wireshark inter-

face, it becomes clear that these messages correspond to commands to reprogram the PLC. Since we did not expect this behavior, we create a new selection of interest only containing these program commands (Figure 3.7b). When selecting the `ip.src` attribute in the Time view and Attribute view, they show that four IP addresses were sending these commands at very specific moments in time over the last 72 hours. Although it is not strange for the main controller to send these commands, the presence of the other IP addresses was unexpected.

3.9. Discussion and limitations

3

The SNAPS approach consists of 5 basic steps: 1) create an overview of potential threats at payload level of a message; 2) use the notion of a lens to inspect alerts in more detail; 3) try to gain insight in the alert by inspecting distributions, temporal patterns and co-occurrence of other alerts in the traffic; 4) create selections of interests to either drill down or analyze traffic from different angles; 5) use close cooperation between machine learning and expert to minimize the number of false positive alerts in the visualization.

Rare values are indicated by the SNAPS classifier, and the additional color rules enable experts to define and reuse insights in suspicious behavior. Another plus is that the reuse of existing visualization techniques and tight integration to the trusted environment Wireshark makes the approach relatively easy to learn. Interaction is kept simple and minimal so that the expert can focus entirely on the traffic data. Views for instance are automatically updated when inspecting values through hovering while the wide range of default selections and the use of auto-completion enables the expert to create/refine selections with minimal effort. The integration between machine learning and visualization makes the system flexible enough to be configured for different environments.

The approach, however, also has some limitations. First, the scalability in the number of attributes and number of selections is limited to the size of the screen. The more attributes that are of interest, the fewer selections can be shown in parallel. Although we provide the expert functionality to hide and scroll between pixel views, this only solves the problem partly. Second, the number of histograms that have to be maintained in parallel linearly increases with the number of selections of interest. For the cases we studied, we found that up to four selections of interest were sufficient for the expert to answer their questions and understand the complexity of their selections. If the number of selections becomes large, however, updating all histograms in parallel becomes too computationally and memory intensive. Third, one disadvantage of the current data acquisition approach is that the quality of the payload analysis highly depends on the dissector. Since the S7 protocol is classified, the Wireshark dissector for S7 was constructed by means of reverse engineering and therefore produces an abstract attribute space. Although we were able to discover some interesting events, the interpretation of alerts in S7 attributes becomes difficult, even with Wireshark.

Finally, some remarks with respect to the classifier. We used a simple and straightforward classifier and will consider alternatives in the future. We used an online classifier, which suffers from producing suboptimal classification results in the presence of traffic bursts. Especially when the data window ω is set too small, traffic bursts can predominate the presence

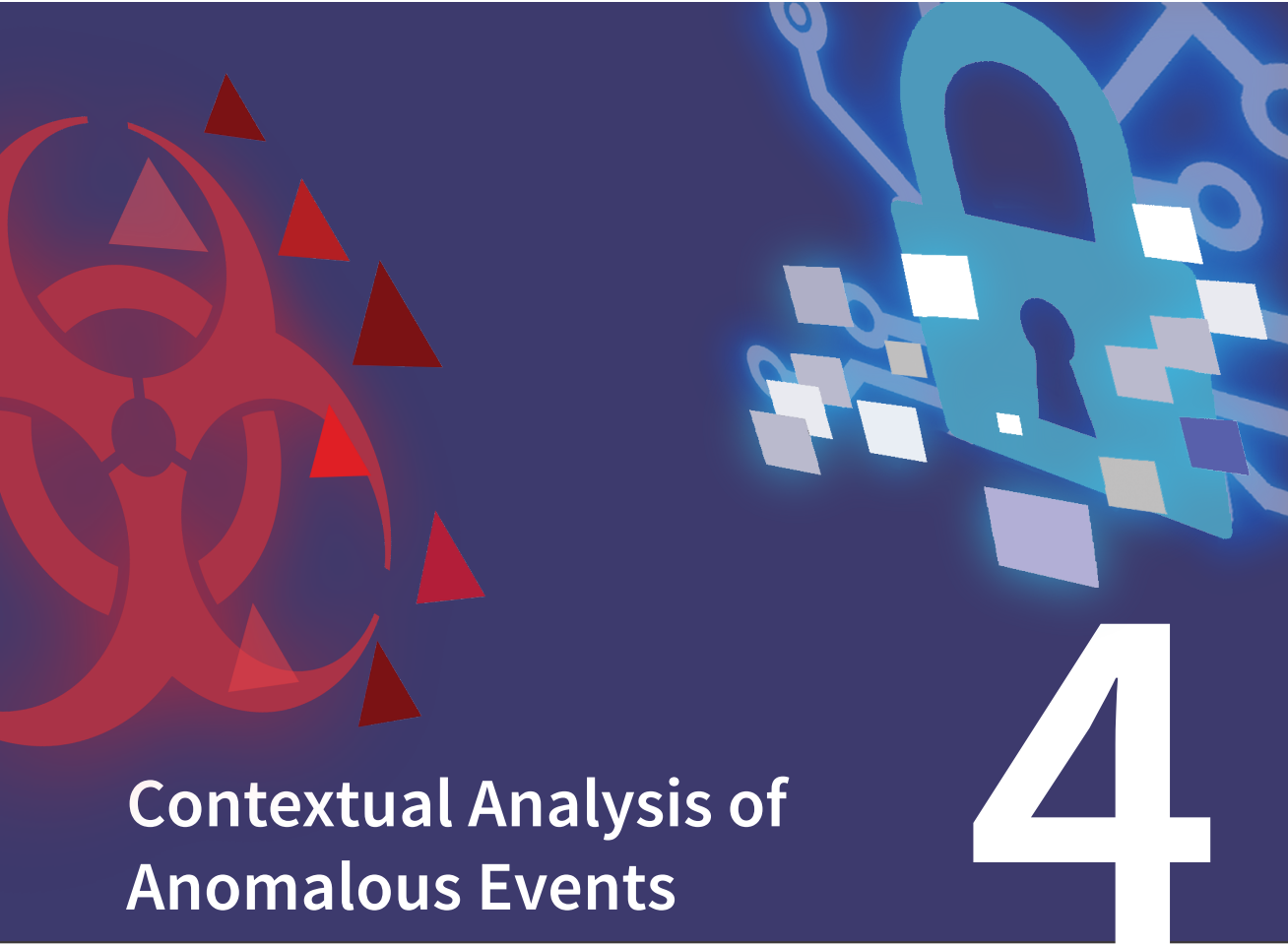
of regular traffic. A partial solution would be to use an offline classifier, but this would require to maintain a separate histogram model for the classifier and data window, thereby significantly increasing the complexity of the approach.

3.10. Conclusions and future work

3

We presented a novel approach for domain experts to discover anomalies in network traffic by combining Deep Packet Inspection, machine learning, and visualization into one coherent system. The ability to create multiple selections in parallel enables the expert to drill down or to focus on specific entities, while still maintaining an overview of the state in the network. The time view enables experts to detect patterns and trends over time, while the pixel, attribute and Lens view together enables the expert to detect outliers. Furthermore, the ability to train and refine classifiers on multiple selections of interest makes the approach flexible enough to be optimized for very specific environments. We have shown the effectiveness of SNAPS on two real-world data sets. Since the approach only relies on the structure of parse data in general, the proposed method is suitable for application in other domains as well.

For future work it is interesting to study how we can analyze network traffic at higher levels of abstraction by grouping messages based on context and structure. This would enable the expert to discover more complex collective anomalies such as file scans or replay attacks. Furthermore, there is still an open question about how the speed of the radars affects the detection rate of the expert. Finally, evaluation is necessary to study the effectiveness and scalability of the approach in larger network environments.



Contextual Analysis of Anomalous Events

4

This Chapter is based on [46]:

B.C.M. Cappers and J.J. van Wijk. Contextual Analysis of Network Traffic Alerts.

In Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec) 2016, pp. 1-8

4.1. Understanding the Context of Network Traffic Alerts

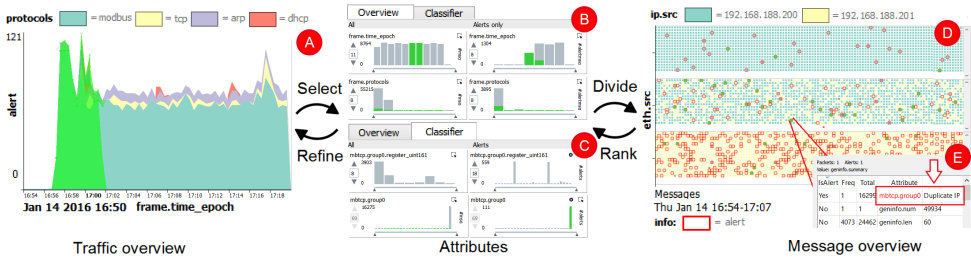


Figure 4.1: The discovery of Man-in-the-Middle behavior in network traffic metadata using selection-based attribute ranking.

4

For the protection of critical infrastructures against complex virus attacks, automated network traffic analysis, and Deep Packet Inspection are unavoidable. However, even with the use of network intrusion detection systems, the number of alerts is still too large to analyze manually. In addition, the discovery of domain-specific multi-stage viruses (e.g., Advanced Persistent Threats) is typically not captured by a single alert. The result is that security experts are overloaded with low-level technical alerts where they must look for the presence of an APT. In this chapter we propose an alert-oriented visual analytics approach for the exploration of network traffic content in multiple contexts. In our approach CoNTA (Contextual analysis of Network Traffic Alerts), experts are supported to discover threats in large alert collections through interactive exploration using selections and attributes of interest. Tight integration between machine learning and visualization enables experts to quickly drill down into the alert collection and report false alerts back to the intrusion detection system. Finally, we show the effectiveness of the approach by applying it to real-world and artificial data sets.

4.2. Introduction

The aim of network forensics is to discover malicious activity inside logs of network traffic. Especially for critical infrastructures, such as power plants, the presence of malicious activity can lead to the malfunction or even destruction of the underlying system. Forensics can no longer limit their analysis to high-level message properties (e.g., length, destination) due to the existence of Advanced Persistent Threats (APTs) [280]. These complex viruses are designed to hide their malicious activity inside the content of messages thereby making them invisible to current flow-based techniques [197].

Since manual inspection of network traffic is impossible due to size and complexity, forensic experts use network Intrusion Detection Systems (IDS) to assist them in finding areas of interest. Although these systems automate the analysis of network traffic, the number of (false) alerts is often too large to analyze one by one. Given that alerts in message content analysis can arise at any combination of hundreds of message attributes, the number of alert types greatly varies. In this chapter we propose an exploration method that enables experts

to gain insight in traffic content by visually exploring and correlating network traffic alerts defined at message-level.

Similar to Livnat et al. [203], we believe that an alert as a result of a complex attack does not stand on its own. The true severity of an alert cannot be determined by solely inspecting its structural properties such as what, when, or where that alert has occurred in the network. Instead, we are interested whether the occurrence of an alert was implicitly related to (a collection of) messages or alerts that were sent in the past. For this we need to be able to inspect message collections for correlations between message attributes (e.g., inter-attribute analysis) and inspect trends in these attributes over periods of time, (e.g., intra-attribute analysis). To enable the simultaneous exploration of message-level phenomena (e.g., field misuse) and traffic-level phenomena (e.g., bursts), our exploration method focuses on a tight interaction scheme between well-established visualization and machine learning techniques. In summary, our main contributions are:

- a visual analytics approach to network forensics, enabling experts to:
 - explore and analyze network traffic on both attribute and temporal level using alerts as a ground truth, and
 - identify and confirm (visual) correlations between network traffic messages and alerts using selection-based relevance metrics and conversation analysis.
- a data-driven coupling between machine learning and visualization for the detection and refinement of network alerts.

This chapter is structured as follows. First, related work is discussed in Section 4.3. Next, the scope and approach for traffic analysis are discussed in Sections 4.4 and 4.5 respectively. Section 4.6 presents an overview of the system and shows how the exploration method is applied. Sections 4.7 and 4.8 describe two example explorations on real-world and artificial data sets and discuss the limitations of the approach. Conclusions and future work are presented in Section 4.9.

4.3. Related Work

A wide range of visualization techniques have been proposed over the years to explore network traffic. We focus here on the approaches that use alerts as a central element. For a broader overview we refer to the surveys of Shiravi et al. [274] and Attipoe et al. [17].

4.3.1. Alert Visualization

Alert visualizations are designed to gain insight into large alert collections, generated by detection systems such as Snort [21] and Bro [244]. Some examples of well-known visualizations are:

- IDS rainstorm [3] visualizes the severity of Snort intrusion detection alerts by creating a pixel visualization of the IP address space where the alerts reside.

- Snortview [174] visualizes Snort alerts over time according to their type, source, and destination. Glyphs and coloring are used to effectively represent false positives.
- Avisa [275] uses a radial display to visualize the relationship between alert types and hosts. Alerts are visualized as B-Splines from alert type to the corresponding host clustered using edge bundling techniques.

Such methods construct an overview of alert collections by visually encoding their severity, source, and type in a single image. Since these methods only focus on alerts as their data source, their knowledge about the normal traffic is limited, making root-cause analysis on this data very difficult. In addition, the loose coupling between IDS and visualization does not enable experts to report false alerts back to the IDS. We believe that a human in the loop approach [264] is vital for quickly gaining insight and reducing the (false) alerts.

4

Methods that do incorporate normal traffic and interactive machine learning in their exploration process are PixelCarpet [192] and SNAPS [45]. PixelCarpet uses a pixel visualization where log entries are represented as a stack of pixels. The brightness of a pixel is used to denote the frequency of log record values. Tight coupling between machine learning and visualization is achieved by enabling the user to remove records from the data set and adapt the model accordingly. Although the technique assists experts in identifying areas of interest through interactive machine learning, their method does not scale for hundreds of attributes. The SNAPS system uses a pixel visualization to display the full structure of a network message as a horizontal line of pixels. Alerts inside messages are highlighted on a per-attribute basis and can be refined using machine learning. Unfortunately, since the approach is focused on monitoring traffic it can only inspect small fractions of traffic at the same time. This makes it hard to detect attacks over larger periods in time.

4.3.2. Exploration

In order to understand the severity and cause of an alert, investigations are needed. Zhang et al. [352] already showed that in flow-based network investigations interaction and multiple views play an important role. In our CoNTA approach, we show that this paradigm can be extended with machine learning and relevance metrics to enable traffic content analysis for hundreds of attributes.

Two systems closest to our technique with respect to exploration are VisAlert [203] and Ocelot [16]. VisAlert discovers correlation between network IDS alerts by visually mapping alerts according to three attributes, namely *what*, *when*, and *where*. They use a radial layout and semantic zooming to find overlap between alerts at various levels of detail over time. Ocelot improves decision support for cyber analysts in computer networks by hierarchically grouping host machines according to various attributes. Filtering is used to isolate affected machines from healthy parts of the network.

VizAlert and Ocelot analyze alerts at the level of a host, rather than at the level of a message. Since host-based alerts only convey information about the network-level constraints that have been violated (e.g., policy violations, access attempts), finding the messages and values that were responsible for these alerts is difficult. In addition, since both methods do not

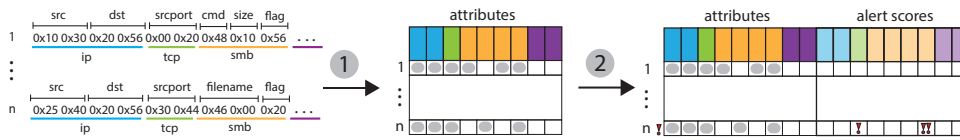


Figure 4.2: 1) Serialization of PCAP traffic with WireShark. 2) Machine learning produces scores per attribute whether these are suspicious or not.

consider the normal events in their decision support analysis, context is lost making it even more difficult to find the root cause of an alert. Finally, both methods do not enable experts to inspect the sequential occurrence of one or more alerts. Kot et al. [178] already indicate that APTs can be the result of a sequence of malicious actions. CoNTA enables experts to visually inspect traffic sequences by inspecting message attributes at the level of network conversations. Interaction enables experts to store and inspect search results in different contexts.

In summary, current methodologies focus on the visualization of large alert collections rather than trying to investigate them through iterative refinement and correlation discovery. The methods that try to discover correlations between alerts and traffic events, either cannot report their findings back to the IDS or limit their analysis to only a few flow-based attributes. Furthermore, their inability to inspect sequential patterns and conversations in normal traffic does not give experts a baseline for determining the severity of an alert.

4.4. Problem statement

The exploration and analysis of network traffic is still a challenge. Even for relatively small networks consisting of tens of nodes, the number of messages per day can easily run in the order of thousands. In addition, every message stores multivariate data depending on its type and purpose. In order to protect environments from APTs, network traffic content has to be analyzed. APTs tend to work in three stages, namely *infiltration*, *expansion* and *exploitation*. Infiltration is typically achieved through social engineering [286]. During expansion, the threat will try to locate the target machine. The exploitation phase is used to sabotage the system by relying on system vulnerabilities. Since APTs exploit domain-specific properties of the network, infiltration is nearly impossible to prevent. We can however detect signs of the other phases by analyzing unencrypted network traffic between hosts for anomalies.

We focus on potential APTs against assets in industrial control systems and office networks, since the modification of assets (e.g., hardware and files) in these networks can have severe consequences. We can identify two types of threats against these assets, namely *system-related* versus *process-related* threats [129]. System-related threats create malicious traffic at network-level and are typically caused by traditional attacks such as buffer overflows and data tampering, possibly assisted by port scans or complex Man In The Middle behavior (MITM) [216]. Process-related threats generate traffic that is legitimate at network-level, but malicious at the level of assets. Examples are raising the temperature of a heater to a 1000 degrees or unauthorized file access. System-related threats are often used as a first step

in the realization of a process-related threat. Process-related attacks are typically the result of an APT.

For the detection of process-related threats, we analyzed application-level protocols SMB2 and ModBus. The Modbus protocol is designed to transfer low-level hardware commands in industrial control systems, whereas SMB2 is used for file management in office networks. System-related attacks can be detected by analyzing flow-based protocols ETH, TCP, and IP.

4.4.1. Data acquisition

4

CoNTA applies semantic network traffic analysis by enriching raw network packets with protocol semantics using WireShark [63]. The result is a multivariate table where rows correspond to messages and columns to attributes. Protocol attributes can represent numerical ranges (e.g., port numbers), strings (e.g., IP addresses), or boolean values (e.g., flag data). Depending on the type of message, specific attributes are present. Since the number of possible protocol attributes outnumbers the number of possible attributes in a message (order of 100s vs. 10s), the resulting table is quite sparse (Figure 4.2).

For the classification of anomalous behavior in the traffic, we aim for an anomaly detection approach, since APT activity is typically not captured by existing signature-based approaches [83]. Similar to SNAPS, we use a probabilistic model where value distributions of protocol fields are learned and decision thresholds are set accordingly [349]. The resulting alerts are used as a basis for the analysis of network traffic. Without loss of generality, we can model an alert as a weighted vector of message values, where the weights describe the extent to which the IDS considers that value malicious. More formally, let D be our multivariate table with N messages and M attributes.

$$D = \{m_{ij}, i = 1, \dots, N; j = 1, \dots, M\} \quad (4.1)$$

where m_{ij} represents the message value of message i at attribute A_j . Furthermore, let S represent a table of alert data such that:

$$S = \{s_{ij} \in [0, 1], i = 1, \dots, N; j = 1, \dots, M\} \quad (4.2)$$

where s_{ij} represents an IDS *alert score* for value m_{ij} . A message m_i is considered malicious if and only if:

$$\exists j [j = 1, \dots, M | s_{ij} > \tau] \quad (4.3)$$

where τ is a decision threshold that for the sake of simplicity is set to 0.95. This model enables us to define classifier refinement techniques that are independent of the underlying machine learning (see Section 4.5.6). Classifiers that do not directly support probability-based classification results can obtain these through posterior probability estimation [171].

Network messages are classified using a probabilistic based IDS for industrial control systems as defined by Yüksel et al. [349]. This classification technique maintains histograms on a per-attribute basis and uses dynamic thresholds to determine the severity of an alert.

For the detection of contextual anomalies involving combinations of values, they derive new attributes using domain knowledge and Pearson's Chi-Square test [249] for statistical independence.

4.5. CoNTA

The size and complexity of network traffic data makes digital network forensics a challenging task. Especially when trying to find the root cause of high-level anomalies, the lack of traffic content can severely limit the investigation. To enable experts in discovering anomalies in this data, we aim for a scalable and interactive visual analytics approach that is coherent with the workflow of traditional digital forensics and cyber defense models [72, 358].

We tackle scalability by summarizing network traffic in a configurable table, enabling experts to inspect trends and outliers from various perspectives by splitting the data over multiple rows and columns. Detailed exploration is achieved by storing message selections as contexts and inspecting them with respect to these contexts. To handle the large number of attributes in the data, attributes are represented as scented widgets [333] that can be ranked and filtered according to characteristics in selected message collections. Large alert collections are tackled by reporting false classification results back to machine learning and enabling experts to analyze alerts from the viewpoint of both messages and attributes.

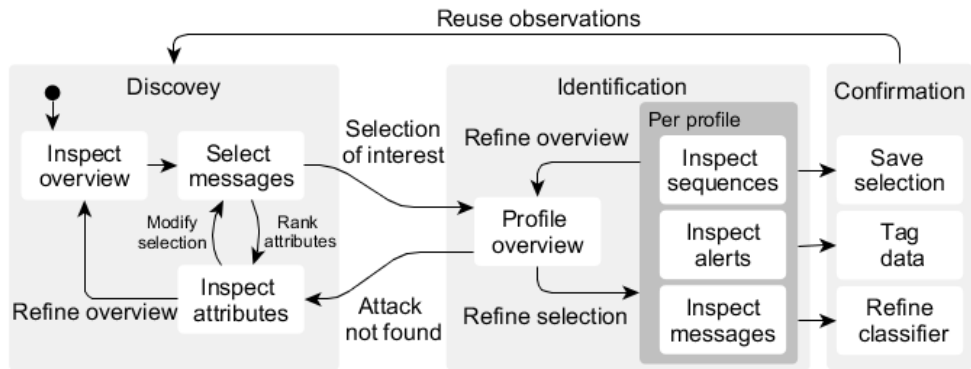
4.5.1. Exploration process

Figure 4.3 shows a schematic overview of the exploration process. Similar to Pollitt's model [251], our approach considers three phases, namely *discovery*, *identification*, and *confirmation*. In the *discovery* phase, experts search the data for areas of interest using alerts as a starting point. In order to determine when a particular subset of alerts is of interest, experts need to find any form of similarity (e.g., originating from the same source, sharing the same attribute) between alerts. For this experts need to be able to:

- compare traffic between multiple entities to discover outliers;
- locate trends and sequences in attributes over time; and
- inspect messages individually to determine their severity.

In CoNTA, experts start their analysis with an overview of the traffic over time on any desired attribute of choice. Suggestions for possible interesting attributes are provided in the attribute view using selection-based ranking techniques (Section 4.5.5).

The *identification* phase tries to locate potential causes of the selected data by splitting the traffic into one or more groups (also referred to as profiles [237]) and inspect them in various contexts. This involves creating hypotheses and verifying them by testing the data for structural properties, such as when and who produced the data and what data were accessed. The inspection of sequential properties enables experts to find malicious message orderings in conversations. This includes the detection of conflicts of interest (e.g., approving your



4

Figure 4.3: The CoNTA workflow model for network traffic exploration. Experts use alerts and a traffic overview to find areas of interest over time in up to three attributes. Selected areas are refined by (de)selecting messages according to suggested attributes of interest. Identification of the underlying problem is achieved by splitting the traffic according to profiles and testing the presence of message values and sequential patterns in multiple contexts. New contexts and attributes are obtained by saving selections of interests. Experts can use this to keep track of their exploration process, compare selections in different contexts, or report false alerts back to the IDS.

own file requests), and violations in operational integrity (e.g., closing the gas valve before lighting a fire).

Confirmation is the phase where conclusions are drawn from the hypotheses. This either results in:

- storing selections into contexts for reuse in investigations;
- tagging traffic with new data to accelerate analysis; and/or
- reporting false alerts back to the machine learning by retraining the data on subsets of the traffic.

CoNTA uses four linked views to assist experts throughout the three exploration phases. In the next sections we discuss the functionality and design decisions for each view separately. For a demonstration of the system in practice, we refer to the supplementary video¹.

4.5.2. Timetable

When analyzing network traffic, the number of messages is typically larger than the number of available pixels on the screen. To provide an overview of the traffic we use a table, where attributes can be inspected over time by grouping the network traffic over at most two attributes of choice. The main motivation for introducing a table of small multiples over a large single is to assist analysts in profiling, where they can inspect traffic with respect to certain attribute values. This enables experts for instance to spot trends or compare traffic over time on a per user or daily basis. For the analysis of the traffic as one large single, the axes of the

¹<https://www.youtube.com/watch?v=yOXDZYKCKZ0>

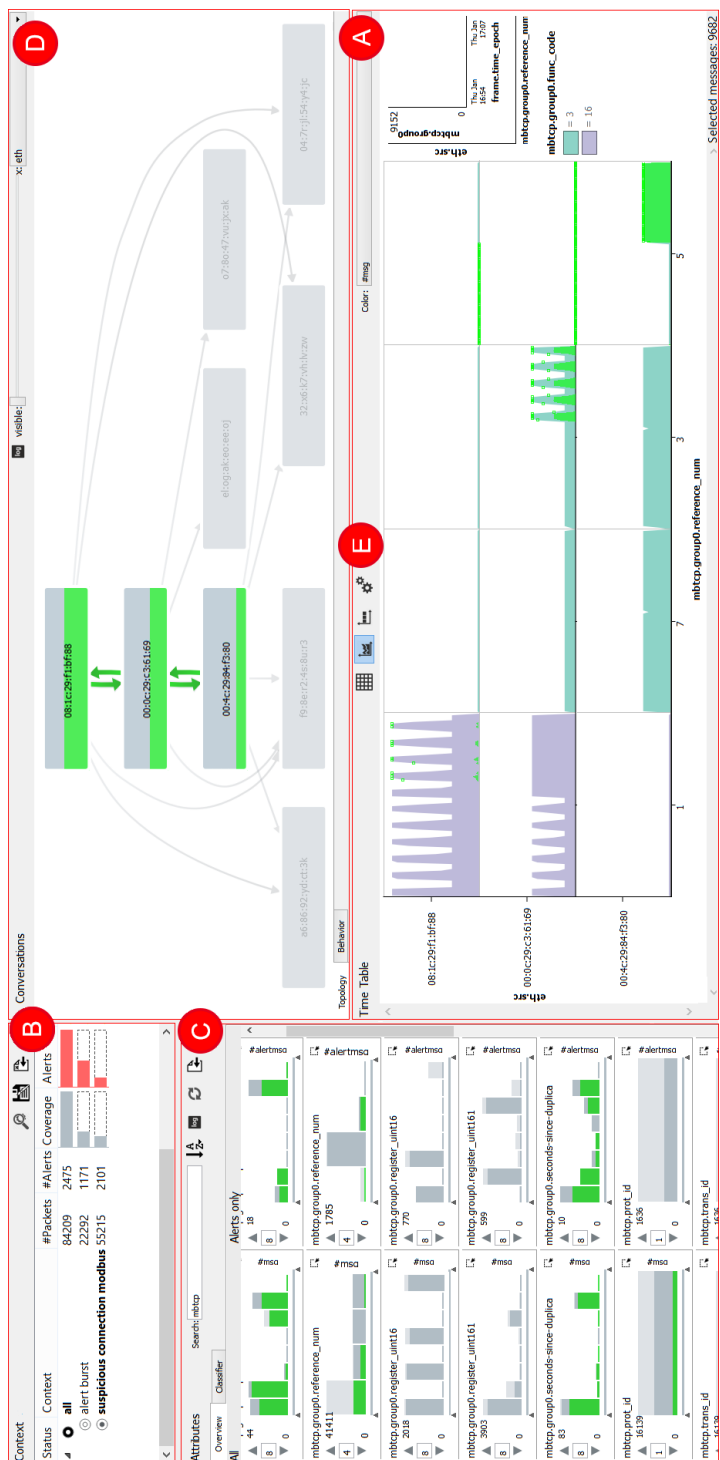


Figure 4.4: Graphical user interface of the implemented prototype and components: a) Timetable for the visual inspection of correlations between 3 attributes over time. Settings with respect to the type of cell visualization (i.e., heatmaps, line charts or pixel maps), axis scaling, and ordering are set using controls in e). b) The context view enables experts to revisit their exploration process and visually compare selections. c) The attribute view shows trends and patterns in selections on a per-attribute basis. Settings with respect to attribute ordering, binning, and classifier settings can be adjusted using the controls surrounding the view. d) Temporal patterns in network conversations can be discovered in the conversation view enabling the expert to inspect the possible presence of malicious messages sequences.

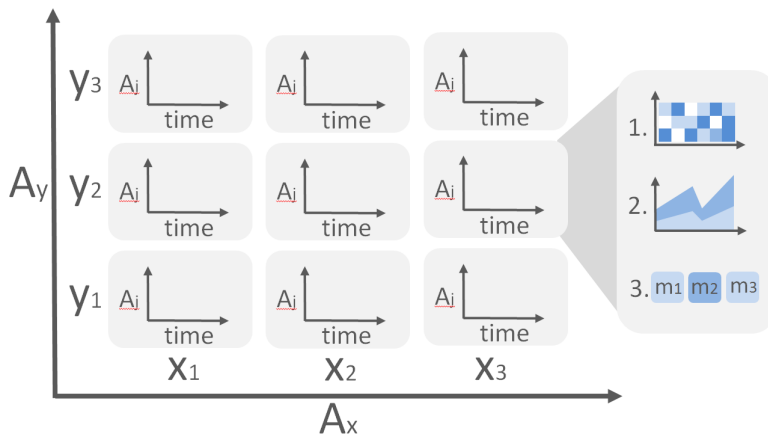


Figure 4.5: The timetable visualizes continuous attribute A_i over time by splitting the traffic over multiple cells using A_x and A_y .

table can be set to `None`. Figure 4.5 shows a schematic overview of how the timetable can be configured. Since repeatedly printing the axis labels for every table cell separately is redundant and can clutter the visualization, a small legend is used instead to inform the expert about the active axes and scaling (Figure 4.4A). For the detection of similarities between one or more table cells, a third attribute of choice can be visualized using color.

The table axes enable experts to analyze categorical and numerical attributes by binning the traffic in non-overlapping intervals. The axes inside each table cell can be used to inspect continuous attributes, such as time or message length. Since time plays a key role in network analysis, the cell's X-axis is always set to time.

Experts can modify the bin sizes of the table and cell axes depending on their task and available space. Small bin sizes are suitable for outlier detection at pixel-level, whereas larger bin sizes can be used for the detection of temporal patterns over larger periods in time. For the detection of patterns between bins, experts can sort axis values by their frequency or rarity. If the number of bins in an attribute becomes too large, experts can enable scroll bars by defining an upperbound on the number of visible bins. Predominating bins can be shown or hidden using the control options (Figure 4.6A).

Table cells

Following from the problem statement, analysts need to be able to compare traffic between profiles, inspect attributes over time, and inspect messages individually. Since every task requires a different perspective, CoNTA supports three visualizations that can be used in the table cells, namely heatmaps, line charts, or pixel maps. The heatmap and line chart are designed to obtain an overview of attributes over the entire traffic. Line charts can be used for inspecting trends, whereas heatmaps enable experts to compare attribute behavior over larger collections of profiles. For the inspection of individual messages, experts can switch

to a pixel map, where every message is represented as a rectangle (Figure 4.1D). By default, messages in the pixel map are colored according to their protocol. Malicious messages are indicated by a red box. Experts can enlarge the pixels through zooming to reveal a summary description of each message. Alternatively, experts can hover over a message to inspect their values by means of a popup (Figure 4.1E).

When selecting a numeric attribute A for the Y-axis of the table cells, the height of every line chart point p is determined by taking the average of all message values with attribute A in p . Attributes `#msg` and `#alerts` are an exception to this rule, since they represent the number of (malicious) messages in each point in time. The color of a heatmap cell is determined according to the chosen colormap. Line charts can also be split based on this colormap to gain insight in the value distribution of a particular attribute over time. This results in a stacked line chart as depicted in Figure 4.1a.

4.5.3. Context view

Experts can save selected messages of interest by assigning a name to them and defining them as a new context. The context view maintains a history of all contexts the expert is interested in. When creating a new context, experts can create a new attribute separating the selected messages from the non-selected. This attribute is added to the data and can be used for further analysis. This enables experts to tag the data with more domain-specific information during exploration.

For every context, the number of messages and malicious messages is displayed. To stay aware of the size of the context, gray and red histograms are used to show the fraction of messages and alerts that are contained in the context with respect to the entire data set. The hierarchy in the view shows the ordering in which the selections were created. Context c is a child of parent context d if and only if c was created when the expert was exploring d . This relationship implies that the messages contained in a child context are always a subset of the messages in the parent.

Multi-context

One danger of drilling down in the data is that overview can be easily lost. Eventually, the fraction of interesting data can become so small that any (visual) significant difference can be misleading due to bad scaling [145]. In CoNTA, experts are enabled to show one context c with respect to an ancestor. The result is that the ancestor context is added transparent in the background of c (Figure 4.4a, c, and d). This enables experts to see any trends and outliers that were currently not incorporated in the current context without losing overview.

We use glyphs in front of the contexts to show when multi-context is enabled. The filled inner circle represents the context of interest (in foreground), whereas the context with the filled outer circle is only visible in the background. Figure 4.4b shows how these glyphs are applied. Experts can enable multi-context by selecting an ancestor context while pressing

Alt. To preserve the parent-child relationship in the context view, experts cannot refine their selection using the messages in the background context.

4.5.4. Conversation view

For the inspection of sequential patterns in conversations, we enable experts to inspect message attributes using a node-link diagram. Let A represent the attribute of interest. The graph is constructed by creating a node for every value in A and there is an edge (v_1, v_2) if and only if a message with $v_1 \in A$ is followed by a message with $v_2 \in A$ in current context c . The thickness of the edges represents the frequency at which values follow each other. Since the resulting graph greatly depends on the chosen attribute and context, we use the general-purpose Dot [113] algorithm for the layout.

4

Note that network traffic consists of multiple conversations running in parallel. Since the order in which conversations are interleaved in the traffic does not have any meaning, by default only sequential patterns within the conversations are being considered. For TCP traffic, a conversation (also known as a session) is defined as the traffic between two IP addresses between two port numbers. In case of numerical attributes, values are binned to reduce the number of nodes in the graph. Experts can prevent nodes with a high degree from occluding the visualization by hiding them using a slider.

Selecting a node v in the conversation view will highlight all messages in the traffic with value v . Selecting an edge (v_1, v_2) selects all message pairs in the conversations where v_1 is indirectly followed by v_2 . A visualization of the network topology can be obtained by creating a graph of all IP or MAC addresses in the network. In contrast to other attributes, both source and destination addresses should be taken into account when constructing this graph. Experts can use this graph to filter the traffic on entire conversations and hosts.

4.5.5. Attribute view

The attribute view shows an overview of all attributes in the traffic using scented widgets. Every attribute is represented as a histogram showing the value distribution of that attribute. The histogram is interactive and can be used to select and deselect messages with specific attribute values. The span slider below every histogram is used to enforce selections within specific value ranges. Timetable axes can be set to a particular attribute through a context menu. The number of bins in a histogram depends on the attribute's distribution. For categorical attributes, there is a bin for every value in that attribute. If the number of categorical values exceeds 20, a miscellaneous bin is introduced to represent the remaining values instead. By default, categorical bins are sorted by their frequency.

For numerical attributes, the number of bins of the histogram is computed using Scott's rule [271]. Experts can modify this number to gain more insight in the outliers of the attribute or to determine the granularity in which experts can interact with the histogram.

Histograms are split into two columns. The left column represents the value distribution of the attributes according to all the traffic in the current context. The right column shows the

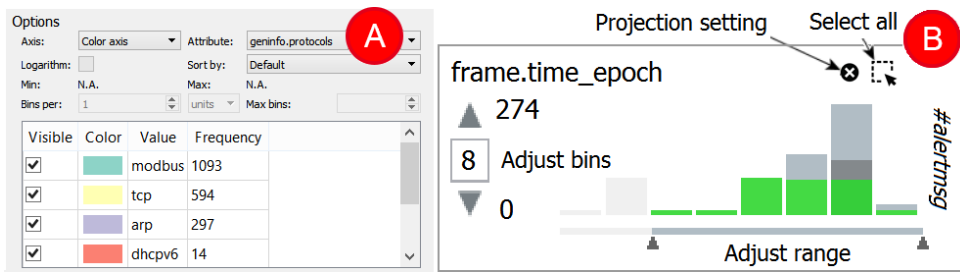


Figure 4.6: A) Options to adjust axis ordering and scaling. B) Overview attribute widget. Projection settings are visible in the classifier tab.

value distribution of the same attribute only considering malicious traffic. The separation enables experts to compare characteristics of malicious messages with the rest of the traffic. In addition, the selection of bins in the right histogram enables experts to search for malicious messages with certain values. More about this in Section 4.6.

Attribute ranking

Since traffic data can consist of hundreds of attributes, inspecting every attribute manually is impractical. Instead, experts can find interesting attributes by sorting them according to various metrics:

- *Alphabetical* sorts the attributes by name.
- *Most Alerts* sorts attributes by counting the number of times a particular attribute was considered malicious in the current selection.
- *Relevance* sorts attributes by computing the information gain [258] for each attribute with respect to the current selected messages. Attributes score high if they can separate selected messages from non-selected messages at best.

Experts can reapply the metrics on specific subsets of attributes by filtering them by name using a textual interface. This is particularly useful when experts are only interested in relationships between attributes within for instance the same protocol.

4.5.6. Classifier integration

False positive rates of network IDSs are still high. To tackle this issue, we enable experts to optimize the underlying classifier through refinement. Since machine learning techniques in IDSs widely vary, we aim for a data-driven approach rather than a classifier-dependent one. In our approach, we support four operations:

- filtering;
- projection;

- binning; and
- self-training [360].

Experts can refine classifier scores by training the IDS on specific subsets of the traffic. With filtering, only messages with values that fit in the specified ranges are included during classification. *Projection* determines which set of attributes should be taken into account during classification. Projection can be used to exclude attributes that are sensitive to false positives (e.g., identifiers).

Yüksel et al. already showed that false positive rates in numeric attributes can be improved by reducing the granularity of the attribute's value distribution through binning [349]. Especially when dealing with values whose alert scores are close to the decision boundary, decreasing the granularity of the distribution can prevent these alerts from happening. Self-training enables experts to report valid messages back to the classifier by labeling them as safe. Instead of ignoring these messages in future classifications, the messages are moved to the training set of the IDS. This enables the IDS to prevent similar alerts in other parts of the data.

In CoNTA, experts can instantly apply filtering, selection, and binning using the attribute view in Section 4.5.5. When switching to the classifier settings, histograms are shown for every attribute in the data set displaying the number of times attribute values were considered malicious. The range slider of a histogram represents the filtering settings, whereas the number of bins shown represents the granularity setting of the classifier for that attribute. Experts can exclude alerts from the projection using the controls in Figure 4.6b. Self-training is achieved through selection and a context menu.

4.6. Interaction

The views in CoNTA work at various levels of abstraction. The line chart, heatmap, and histograms inspect patterns at traffic level, whereas the pixel map and conversation view work at message and conversation level respectively. For the investigation of alerts in CoNTA, linking and interaction play a key role. To ensure that brushing and linking is consistent and understandable over all views, we decided to use messages as a central concept. Message-oriented interaction across views enables experts to reason about their selections as sets of messages. Additional messages can be selected in different views by selecting visual elements while holding the `Ctrl` key (e.g., set union). Deselecting elements will remove the messages from the current selection (e.g., set difference).

To preserve consistency when creating a selection, every visual element (i.e., heatmap cell, histogram bar, line chart series, and graph node/edge) is filled with a green color proportional to the fraction of the selected messages in that item (Figure 4.7). To ensure that the intensity of the item's background color is preserved, transparency is added to the selection color. Similarly, hovering the mouse over an item will show the fraction of hovered items (that were not already selected) as a translucent dark gray color on top of the selection. This enables experts to see the impact of the new selection before applying it. To prevent elements

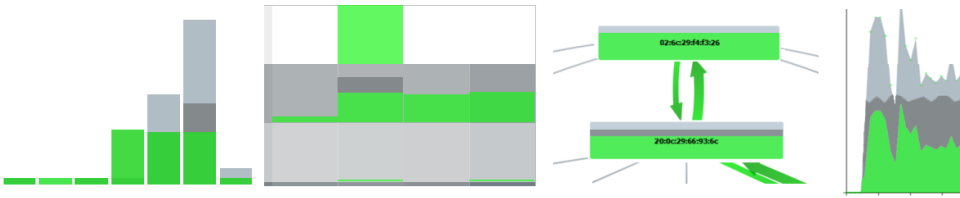


Figure 4.7: Brushing and linking is applied accross all views in CoNTA.

and selections from getting visually too small to properly interact with them, the height of every element and selection is set to a minimum of two pixels.

4.7. Use cases

We tested the effectiveness of our method on one artificial and one real-world data set. The first data set represents the simulation of a fully functional artificial water plant consisting of 5 hosts, 80,000 messages and 170 attributes. The data set was designed by an external security company that is specialized in the detection of malicious activity in industrial control systems. To show the practical existence and impact of APTs, they injected an APT to damage the facility. The second data set is obtained by recording 3 days of internal SMB2 network traffic from a university for which there was no ground truth known beforehand. The data set corresponds to approximately 800,000 messages, 400 attributes, and was sent by approximately 20 hosts. For a better experience of the interaction and use cases in practice, we refer to the supplementary video.

4.7.1. Water plant

Discovery

We initially start exploring the data set by inspecting the number of alerts in the network over time using a line chart. We select the burst period between 16:55 and 17:10 PM in the line chart and save the messages in a new context called “alert burst” (Figure 4.1a). According to the topology of the network, most traffic was created by three nodes: the water tank (... : 80); the SCADA system monitoring the plant (... : 88); and a router in between (... : 69) (Figure 4.4d). Alerts that were caused by infrequent protocols in `frame.protocols` are removed using projection (Figure 4.1b). Alerts that involve rarely active hosts are removed by selecting the infrequent bins in the right `ip.src` histogram of the attribute viewer and reporting them back to the IDS. After selecting the new context, we sort the attribute for most common alerts. The attribute `mbtcp.reg_uint16` scores high indicating that many different messages with strange register values were seen by the IDS. Our eye was caught by the attribute `mbtcp.group0` whose right histogram shows that there are 50 messages with alerts that have the value `duplicate IP` (Figure 4.1e).

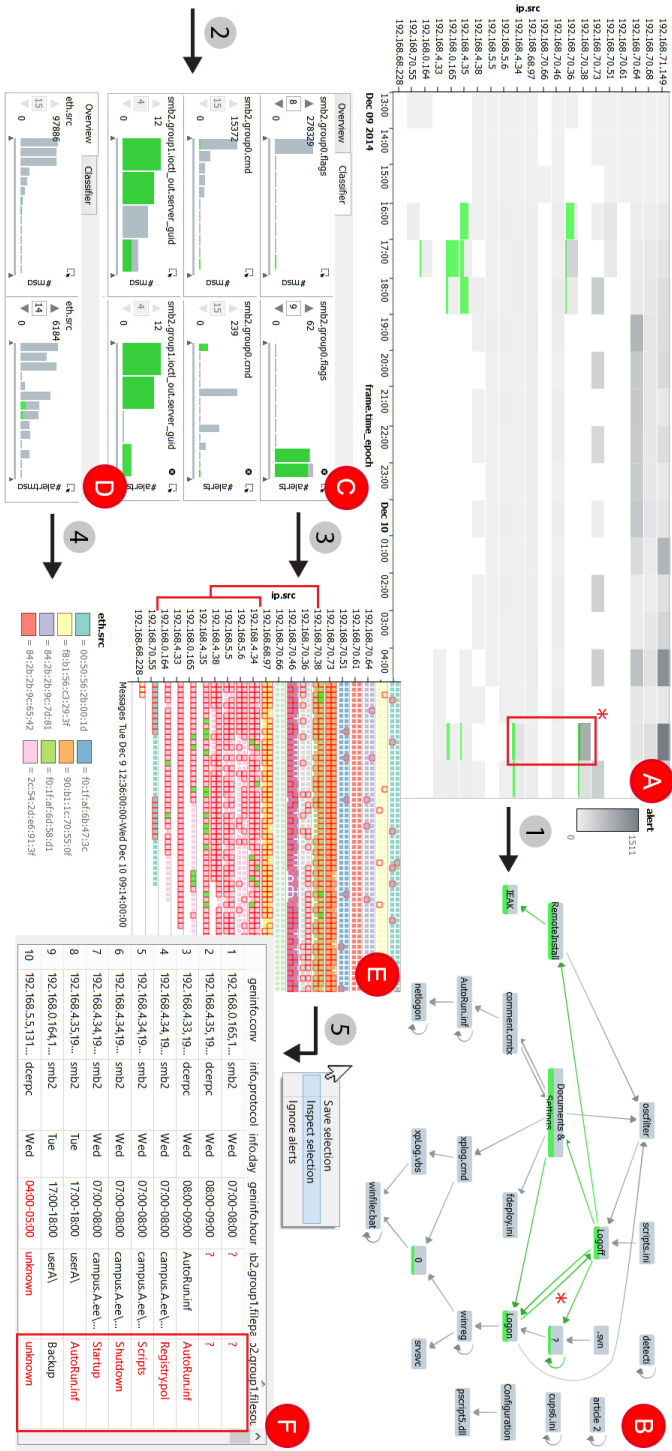


Figure 4.8: a) Heatmap showing the number of alerts per IP address per hour. b) Conversation view on smb2.11enames to detect the presence of strange access orderings. c) Attribute sorting on Most Alerts shows that selected alerts use the same smb2_group0_flags. d) Sorting attributes by relevance after selecting messages with the same flags. e) Pixel map per IP colored by ip_src. f) Tabular view of alert selection.

Identification

We select all alerts with the strange value by switching to the classifier interface using the right histogram in the attribute view (Figure 4.1c). Sorting the selection by relevance shows that most selected messages were received by two MAC-addresses. We group the traffic per MAC-address by setting the timetable's Y-axis to `eth.src`. Switching to the pixel map shows that most alerts are present in the water tank (Figure 4.1d). Coloring the messages by IP source reveals that the router uses the same IP addresses as the other nodes, suggesting the presence of man-in-the-middle activity. We select the conversations between the three nodes using the edges in the conversation view, filter the traffic by Modbus, and create a new context for them for further investigation.

Confirmation

Now that we know that this router is suspicious, the next step is to find out what the router is aiming for. We select all malicious messages that were sent by the router using the right histogram `eth.src` in the attribute viewer. Filtering this view by only Modbus fields and sorting the widgets by relevance reveals that most alerts were caused when reading particular registers (Figure 4.4c). Since each register in the plant stores its own data, we group the traffic per register by setting the timetable X-axis to `mbtcp.ref_num` and table cell Y-axis to `mbtcp.reg_uint16`. Figure 4.4e now shows how register values sent by the water tank are actually perceived by the SCADA system and vice versa. Register 1 shows the status of the tank's valve, where the height of the register value describes the extent to which the valve is open. Register 5 represents the overflow flag the tank raises when the water level exceeds a certain threshold. Note that the close valve commands that are sent to the router are not forwarded to the tank. Further note that the tank's overflow flag is suppressed by the router. The result is that the tank overflows while the SCADA system is unaware of this situation. Finally, we select the `All` context in the background to see in the conversation window that this router apparently also has conversations with other hosts in the network (Figure 4.4d).

4.7.2. University

Next, we show how sequences of messages can assist the expert in investigating when and how files are accessed in the network. We first create a new context only containing SMB2 messages with file names by selecting all bins in the left histogram of the attribute `smb2.fi-`

`resource`. Since the number of hosts in the network is significantly larger compared to the previous data set, we start exploration by creating a heatmap showing the number of messages per IP address (Figure 4.8a). IP addresses in the subnets `*.*.4.*`, `*.*.5.*`, and `*.*.71.*` generate traffic over day and night. Switching to the pixel map and coloring the pixels according to their SMB2 command, shows that these addresses are servers only sending response messages. After sorting the Y-axis by frequency and coloring the heatmap by number of alerts per hour shows that on December 10 06:00-07:00 AM IP addresses `192.168.4.34` and `192.168.70.38` generated more alerts with respect to the

other time slots. Selecting both heatmap cells shows in the conversation view that both hosts were communicating with each other.

We inspect the file access behavior of the hosts by creating a node-link diagram of all the files that were accessed in the network. Frequent files that are accessed by everyone (e.g., the file `spoolss` whenever a document is printed) are hidden using the visibility slider. Besides the strange ? in the graph, we see that a `Login` file is indirectly followed by a `Logoff`. Selecting the edge from `Login` to `Logoff` shows that this behavior is generated by `192.168.4.34` and `192.168.70.38` (Figure 4.8a, red box). Selecting the suspicious heatmap cells shows in the graph that these hosts accessed a wide variety of files within an hour such as `Autorun.inf`, `scripts.ini`, and `RemoteInstall` (Figure 4.8b). Sorting the attribute view on *Most alerts* shows that most of the selected messages have the `smb2.group0.flag` set to 8 instead of 0 (Figure 4.8c). Deselecting the other flag values and sorting the attributes by relevance shows that these alerts were only generated by two MAC addresses (Figure 4.8d). We switch back to the pixel map and color the messages by MAC address. This shows that one user runs multiple virtual machines on the same host (Figure 4.8e). The interesting part however is that the servers we detected with subnets `*.*.4.*` and `*.*.5.*` are all originating from the same machine. Inspecting the alerts that were generated by these addresses using a table view, we can see that the machine was accessing rather interesting file names in the network (Figure 4.8f).

4

4.8. Discussion and limitations

The use cases in Section 4.7 illustrate that there is a strong interplay between high-level traffic overviews, low-level message views, and attributes. The tight linking between the different views plays a key role in understanding how high-level phenomena such as bursts relate to the presence of low-level alerts in messages. By tagging message collections through (de)selection, network traffic can be incrementally enriched with intuitive domain-specific descriptions.

The definition of an outlier greatly depends on the domain knowledge and the context in which the data is observed. The access of a file `X` does for instance not have to be malicious in general, but can be dangerous when performed by a certain user. The exploration method should therefore be flexible and expressive enough to create and inspect new selections without much effort. The timetable facilitates this by enabling experts to inspect traffic with visualizations they are familiar with. Combined with multi-context functionality, outliers can be inspected in various contexts with a single mouse click. Being able to select and deselect messages based on their attributes, values, and temporal occurrence in conversations, while directly gaining feedback on both message and traffic level provides experts with a powerful exploration mechanism.

Like any methodology, there are limitations. First, the number of small multiples in the timetable does not scale well when considering attributes with many different values. Although the expert is enabled to hide values and use scroll bars to limit the number of displayed values, this only solves the problem partly. Furthermore, the node-link diagram in

the conversation view does not scale when visualizing large networks. Note however that the analysis of alerts hardly involves the analysis of all the traffic in the network at once. Since the analysis of alerts quickly narrows the area of interest, we decided to choose visualization methods based on their understandability and commonality, rather than their scalability.

Second, the interaction with attributes is limited to the number of visible scented widgets. Showing too many attributes will break the interaction whereas too few attributes will increase the risk of missing potential correlations. Although sorting, filtering, and scrolling helps to find interesting attributes, creating queries involving many attributes can become a burden and a textual interface is preferred.

Third, the proposed classifier refinement approach implicitly assumes that the underlying classification model is suitable for semi-supervised learning. Although the interaction enables experts to train the classifier additionally on specific parts of the traffic, there is no clear boundary between fitting and overfitting the underlying model. The extent to which an expert can detect a false positive can greatly influence the classifier's performance in a good or bad way.

4.9. Conclusions and future work

We presented a novel approach for domain experts to explore large message collections using automatic generated alerts and interaction. The ability to interactively switch from traffic-level overviews to message-level details enables experts to investigate the relationship between high-level traffic phenomena and low-level message fields while staying aware of other concepts such as conversations and sequential patterns. The combination of attribute-based scented widgets and selection-based relevance metrics enables experts to search through large attribute collections and refine classification results in multiple dimensions. Since the methodology exhibits the structure of time-dependent multivariate data, it is general and flexible enough to be applied in other domains. We have shown the effectiveness of the approach on real-world and artificial data sets.

For future work it is interesting to see how we can enrich our method by training new classifiers on specific subparts of the traffic. This would enable experts to interactively test the IDS performance for different types of profiles. Developing an intuitive interaction mechanism, however, is nontrivial. Furthermore, an extensive evaluation is required to study the approach's real-time capabilities and effectiveness in different network environments and domains.



Exploration of Multivariate Collective anomalies

5

This Chapter is based on [47]:

B.C.M. Cappers and J.J. van Wijk. Exploring Multivariate Event Sequences using Rules Aggregations and Selections.
In IEEE Transactions on Visualization and Computer Graphics 2018, 24, 1, 532-541

5.1. Exploring Event Sequences using Rules, Aggregations, and Selections

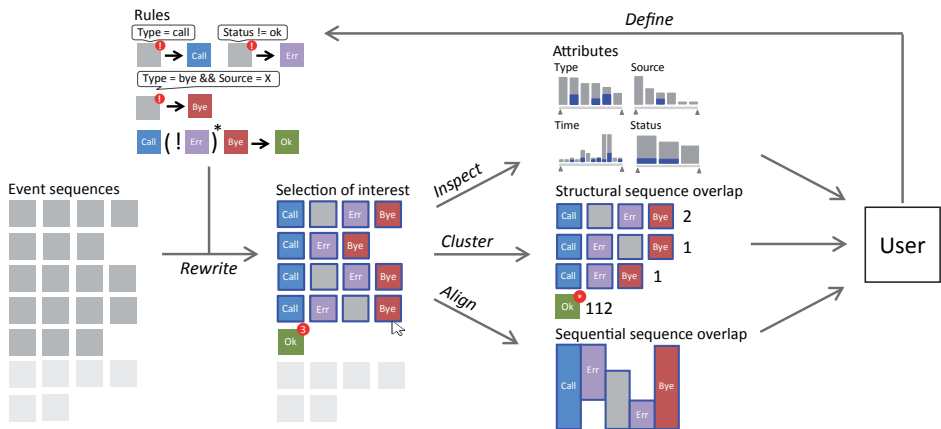


Figure 5.1: The Eventpad dataflow model for multivariate event sequence exploration: Users construct rules to visually encode events of interest using attributes and domain knowledge. Exploration of rewritten events is achieved by inspecting overlap in multivariate data of selections of interest. Discovery of structural overlap between sequences is achieved by clustering sequences based on their visual encoding. Overlap between sequentially different sequences is discovered using multiple sequence alignment. New insights can directly be incorporated in the analysis by storing selections of interest and defining new rules throughout exploration.

Multivariate event sequences are ubiquitous: travel history, telecommunication conversations, and server logs are some examples. Besides standard properties such as type and timestamp, events often have other associated multivariate data. Current exploration and analysis methods either focus on the temporal analysis of a single attribute or the structural analysis of the multivariate data only. We present an approach where users can explore event sequences at multivariate and sequential level simultaneously by interactively defining a set of rewrite rules using multidimensional regular expressions. Users can store resulting patterns as new types of events or attributes to interactively enrich or simplify event sequences for further investigation. In Eventpad we provide a bottom-up glyph-oriented approach for multivariate event sequence analysis by searching, clustering, and aligning them according to newly defined domain-specific properties. We illustrate the effectiveness of our approach with real-world data sets including telecommunication traffic and hospital treatments.

5.2. Introduction

Many domains nowadays try to gain insight in complex phenomena by logging their behavior. Telecom companies for instance analyze their communication networks for the presence of fraud, hospitals analyze patient treatments to discover bottlenecks in the process, and companies study their workflows to improve customer satisfaction. The common ground here is that domains are interested in the analysis of *sequences* (e.g., phone calls, treatments,

workflows) in their system by recording *events*. Without loss of generality, we define a *sequence* (a.k.a. trace, record, session, case, or conversation) as a series of events that have the same *sequence_id*. Besides their type and temporal information, events often have more associated information (e.g., status code, source, length) depending on the domain. In addition, the number of events in real-world data is typically in the order of millions and more.

Multivariate event sequence exploration is still a challenge due to size and variety. Current methods often limit the analysis of event sequences to a single attribute without considering other multivariate event properties in the definition of patterns. We believe however that for root-cause analysis of anomalous sequences the two should be explored simultaneously, since values in multivariate data are often crucial to understand patterns in sequences and vice versa. For example, although sequences of requesting, accessing, and modifying a file in general are not suspicious, they can be considered malicious when they are requested by a particular group of users, the type of file is invalid, and/or one or more authentication errors occurred throughout the execution. For this we need to be able to incorporate multivariate data of events and sequences in our sequential analysis. The observation for instance that all request events in anomalous sequences share particular characteristics in one or more attributes can help analysts to gain insight in the underlying problem.

To enable simultaneous exploration of attributes and event sequences, our exploration method focuses on a user-oriented approach where the inspection of selections of interest, creation of rules and reinterpretation of event sequences according to these rules play a central role. To support this flexibility we introduce Eventpad, a notepad editor for multivariate event data. More specifically, our main contributions are:

- an exploration method that enables users to simultaneously explore and analyze multivariate event sequences on both multivariate data and sequential level, using
- a glyph-oriented visual query interface to define higher-level patterns using *multivariate* regular expressions,
- a uniform interaction scheme for the creation and modification of selections of interest across events and attributes, and
- a tightly coupled dual view for the discovery of overlap and anomalies between event sequences through interactive multiple sequence alignment and sequence reordering.

This chapter is structured as follows. First, related work is discussed in Section 5.3. Next, we describe our data model and approach to multivariate event sequence exploration in Section 5.4. Sections 5.5, 5.6, and 5.7 present an overview of the system and discusses design decisions with respect to visualization, data manipulation, and interaction. In Sections 5.8 and 5.9 we provide two example explorations on real-world data sets and discuss the limitations of the approach. Finally, conclusions and future work are presented in Section 5.10.

5.3. Related Work

Event sequence exploration is an extensively studied topic covering a wide range of visualization techniques. The most well-known method to visualize event sequences is by representing events as glyphs based on their type and point (or interval) in time [41, 82, 85, 174]. To enable exploration in event sequence analysis across different tasks [33], most systems enable users to perform operations on events and sequences such as grouping, aligning, searching, sorting, and filtering [336].

Event sequence exploration methods can be grouped into two main categories: exploration through *overview* or through pattern *searching*.

5.3.1. Event Overview

5

An overview is often obtained by showing events using icicle plots [24, 182, 338], state diagrams [337], pixel maps [46], “piano roll” glyph displays [85], or networks [245]. As opposed to the visualization of all event sequences, other approaches focus on the visualization of similarities [209] or differences between sequences, such as MatrixWave[356]. In order to cope with large data volumes, many overview techniques provide filtering and clustering capabilities to reduce visual complexity.

A common approach for event visualizations is to visually encode events according to their type. In case of a large number of event types, this encoding is limited to the most frequent types in the data. Event types are important, but for detailed analysis often multiple attributes have to be taken into account to highlight relevant events. Hence, we aim for an alternative approach where we, similar to Tominski [305] and Bernard et al. [25], enable users to visually annotate and simplify [156, 184] the data that at that moment are relevant for their analysis. To support this we enable users to specify *rules*.

5.3.2. Event Searching

Various visual *search* interfaces have been developed for the construction of time-interval based queries [94, 225], regular expressions [18], analysis of cohort selections [181], and web session logs [189]. A query interface closest to Eventpad is (s|q)eries [351]. The (s|q)eries system enables users to visually construct regular expressions on multivariate data associated with events. Users can drag and drop multivariate constraints as blocks in a node-link diagram to build their query of interest. More complex regular expressions can be obtained by adding operators to the blocks and connecting them sequentially via edges. (s|q)eries is a query-driven system that is effective for finding known patterns of interest. For the specification of queries, however, users need to be well aware of the available attribute space, since (besides searching) there is no other support for event sequence exploration and attribute analysis.

5.3.3. Event Exploration

An exploration method both supporting overview and search mechanisms and close to our technique is EventFlow [223]. EventFlow is an extensive tool for the exploration and analysis of large event collections. Custom interfaces are provided to intuitively search, filter, categorize, align, and simplify [224] events based on type, timestamps, and time intervals. Although the system provides extensive support for event analysis by their type and temporal information, Monroe et al. indicate that the system provides little support for the analysis of additional multivariate data [223]. In addition, they also indicate that comprehensive support across event sequences and attributes is key for temporal event sequence analysis tools to be used in practice. Other techniques try to overcome the problem of multivariate analysis by considering the attributes together as one event type [122]. This approach however does not scale well for high-dimensional data.

Another system for the simultaneous analysis of temporal patterns and multivariate data is ClickStreamVis by Liu et al. [202]. ClickStreamVis enables the analysis of multivariate data in event sequences by extracting frequent sequential patterns from the data using pattern mining techniques. As opposed to existing techniques, ClickStreamVis tries to obtain higher granularity levels by automatically extracting (sub)sequences of interest using motif analysis and various clustering techniques. Unique event sequences are visualized in an icicle plot along with their frequencies. Analysts can align sequences in the visualization according to events in the mined patterns to discover areas of interest. Liu et al. already indicated that their sequence view does not scale well due to the wide variety in large event sequence logs and lack of semantic zooming in their tooling. Furthermore, the computation of mining and pruning maximal sequence patterns can take minutes for relatively small data sets. In Eventpad we tackle wide diversity between event sequences by enabling users to filter or simplify sequences using regular expressions [172]. In addition, users are enabled to store intermediate selections of interest for further investigation. Since the evaluation of a regular expression is linear in the size of the input and its corresponding deterministic finite automaton [4], the technique easily scales to the analysis of hundreds of thousands of events.

In summary, current methods either focus on the sequential analysis of univariate data or the structural analysis of multivariate data. Systems that do take multivariate properties into account during analysis either focus on obtaining an overview or providing search capabilities to explore the data. Currently, no system provides users the ability to both explore and search event sequences by combining temporal patterns and multivariate data in one interface and query language. In addition, no system enables users to interactively explore overlap between sequences of interest alongside attributes using multiple sequence alignment and rule-based event rewriting.

5.4. Exploration

The size of and variety in large event sequence logs makes it difficult to gain insight in the behavior of the underlying system. Users are not only interested in the existence of particular sequences, but also want to understand what might have caused them. In order to do so, they

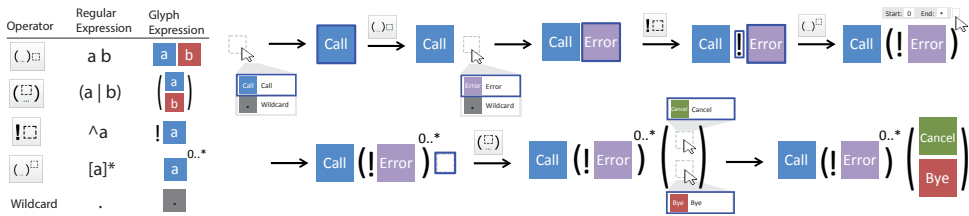


Figure 5.2: Incremental construction of regular expressions with predicate logic. The example query involves two different attributes and returns all phone calls starting with a CALL and ending with a BYE or CANCEL without 0 or more error messages in between. Operators can be applied over multiple glyphs by first selecting them through range selection.

need to be able to discover which multivariate event properties in these sequences distinguish them from the rest. Since higher level concepts such as a “bad login attempt” or “successful phone call” are typically not captured in low-level events, users need to be enabled to enrich their data with these concepts to find the sequences they are interested in. In summary, we need a scalable interactive method to

- inspect event sequence orderings of interest alongside their associated multivariate data,
- assist users in finding and defining sequences of interest, while
- staying aware of high-level phenomena in sequence collections through overview.

In order to keep this method simple and scalable for the analysis of large event logs, we choose for a bottom-up exploration approach where event sequences are represented by series of glyphs and high-level overviews are obtained by finding and summarizing sequences based on user-defined patterns of interest. To tackle the variety in event logs, we consider the problem at three levels, namely the reduction of complexity:

- within event sequences,
- between event sequences, and
- inside multivariate data (of events of interest).

For this we use three concepts, namely

- rules,
- pattern aggregation, and
- selections.

Rules enable users to simplify and visually encode event sequences using glyphs and regular expressions. Users can apply *pattern aggregations* to discover overlap between *sequentially* similar but *structurally* different sequences by summarizing them using clustering and alignment techniques. The creation of event *selections* of interest enables users to focus on parts of the event sequences that are relevant for their investigation.

Figure 5.1 shows a schematic overview of how these concepts are used in Eventpad. In terms of the strategy guidelines of Du et al. [82], Eventpad enables users to extract records and categories (S1,S2), identify features of sequences (S3) or alignments (S4) with the ability to group events by custom defined categories (S9) and coalescing event sequences into new events (S10, S12). Newly defined search patterns can be stored and applied to larger data sets (S14). For a demonstration of the method in practice, we refer to the supplementary video¹.

5.5. Rules

Text editors typically enable text filtering, highlighting, and compression by providing search and replace functionality through *regular expressions*. Regular expressions are the de facto standard in industry systems such as Elasticsearch [121], Logstash [311], or `grep` [168] for efficiently finding (and replacing) character sequences in text according to search patterns. Traditional regular expression languages only operate on univariate data such as plain text. We describe how we enable multivariate event sequence analysis by extending regular expressions with predicate logic to support multiple attributes.

5.5.1. Formal theory

A traditional regular expression consists of *symbols* (i.e., text) and *operators*. In order to extend regular expressions to support multivariate data, we define two types of events, namely *micro* and *macro* events.

A *microEvent* e has (attribute,value) pairs $(a, v) \in A \times V$ where v can represent numerical ranges, strings, or boolean values. In addition, a *microEvent* has a `sequence_id`. We model a *macroEvent* $e' = \langle L, ES \rangle$ where L is a list of labels and ES a set of *microEvents*. Initially we assume that every *microEvent* e is contained in a (default) *macroEvent* e' with $L = ["Gray"]$ and $ES = \{e\}$. We can now model an event sequence as a series of *macroEvents* with the same `sequence_id`.

Our extended regular expression language consists of *macroEventPredicates* and *operators*. A *macroEventPredicate* is a boolean expression B over label(s) in \mathcal{L} and/or attribute(s) and value(s) in A and V . A *macroEvent* m satisfies B if and only if **all** *microEvents* in m satisfy B . Alternatively, one can require **at least one** *microEvent* to satisfy B . We refer to this as *maximal* versus *minimal* matching respectively.

A rule is of the form $\alpha \rightarrow l$ where α is a regular expression and l is a label. Operationally, a rule is fired if an event sequence s in the data set can match α . This results in the replacement of s by a new *macroEvent* $c = \langle L', ES' \rangle$ where $L' = [l]$ and ES' is the union all *microEvents* in s . In case of one-to-one mappings, labels in s are prepended to L' . This enables users to reason about multiple labels when specifying queries (more about this in Section 5.5.3.)

¹<https://www.youtube.com/watch?v=2DWVW-vLN8Q>

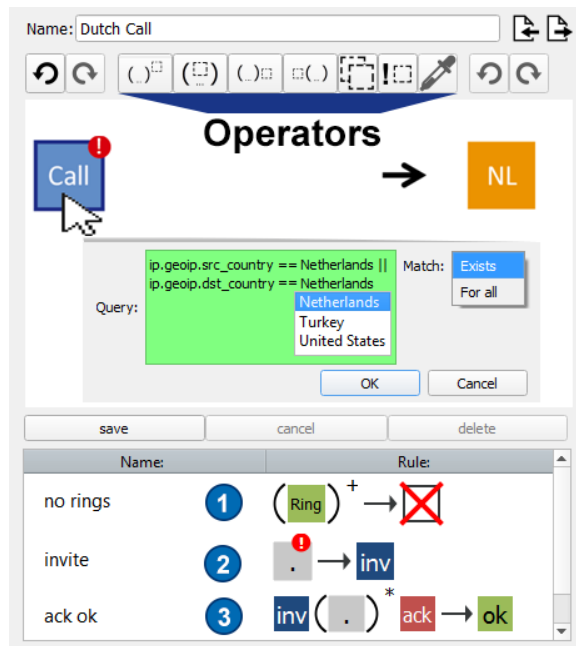


Figure 5.3: Query interface for construction of rules. Boolean constraints can be added to a glyph using a textual interface with autocompletion and query validity checks. The pipet enables users to copy/paste glyphs that are present in the sequence view in the query interface. Glyphs with an exclamation mark have one or more attribute constraints.

In contrast to traditional regex, we have to ensure that events before and after the application of a rule remain the same. To keep this mapping simple and intuitive, we limit a rule to the creation of one macroEvent. For more details, we refer to the supplementary material.

5.5.2. Rule visualization

Initially, every macroEvent is represented by a gray glyph.

The sequence view (Figure 5.4A) visualizes event sequences as a series of glyphs, starting every sequence on a new row. Event sequences that do not fit on a single line are wrapped over multiple rows. Scroll bars are used to inspect the entire data set. The size of the glyphs can be set proportional to a numeric attribute of choice. Since Eventpad focuses on the reduction and analysis of event patterns, time-intervals between events are disregarded in the visualization.

Users can replace one or more macroEvents in sequences by a new one using a rule editor (Figure 5.3). Traditional regex operators such as sequential composition, choice, and iteration (0 or more times) can be used to construct patterns. Figure 5.2 shows how these operators are visually encoded in the interface. Similar to Word's equation editor [217], operators can be combined to construct more complex patterns.

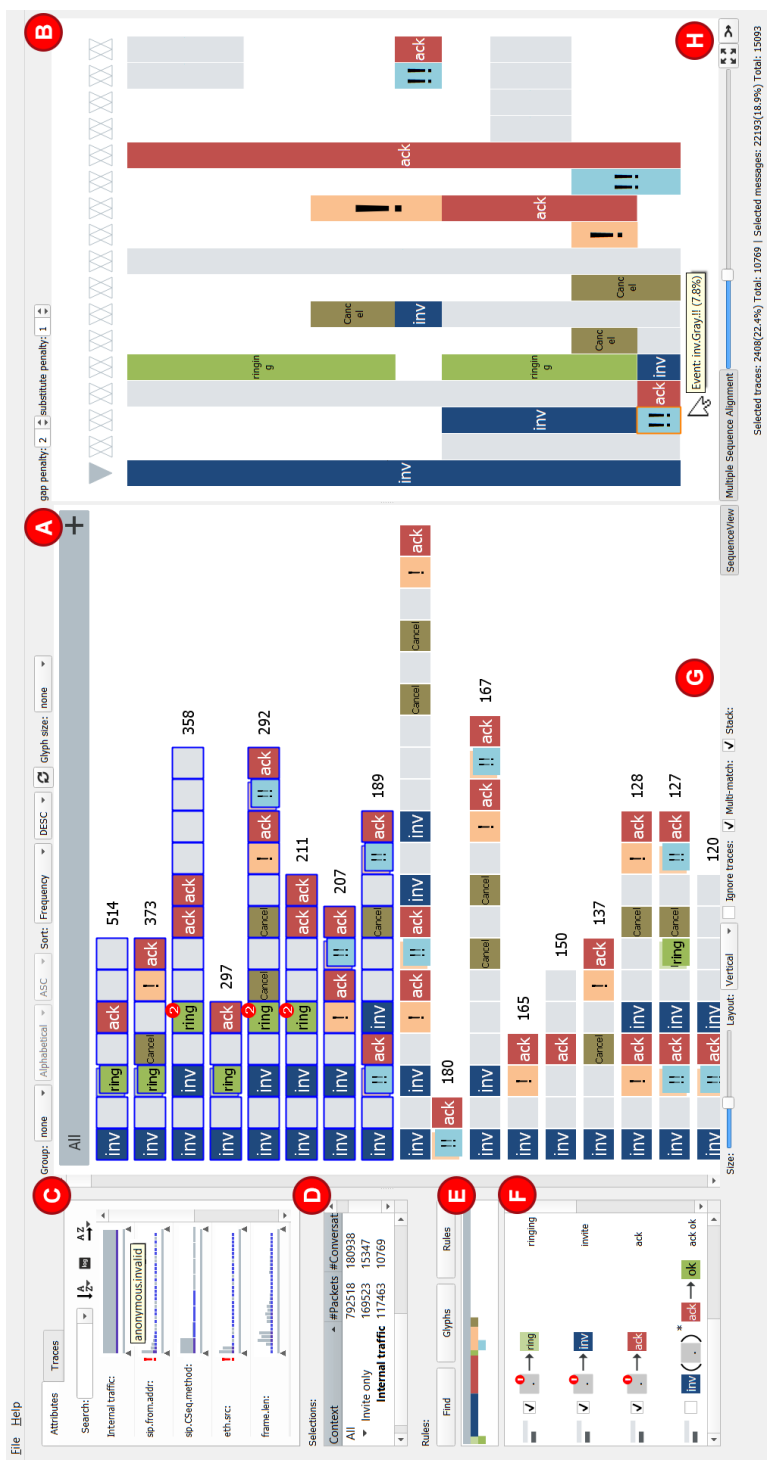


Figure 5.4: Graphical user interface of the implemented prototype and components: A) Sequence view represents sequences as a series of glyphs. Settings with respect to sorting, grouping, and clustering of sequences are set using controls in A) and G). B) The alignment view finds overlap in sequences of interest by aligning them. Alignment parameters, layout, and sort settings can be modified in B) and H). C) The attribute view shows trends and patterns in selections on a per-attribute basis. D) Context view enables experts to store selections of interest throughout exploration. E) Rule overview widget to show the coverage of applied rules. F) Rule view shows a list of applied rules along with their settings. The ordering for rewriting is controlled via drag and drop operations. Rules can be toggled on or off along with longest and shortest match settings.

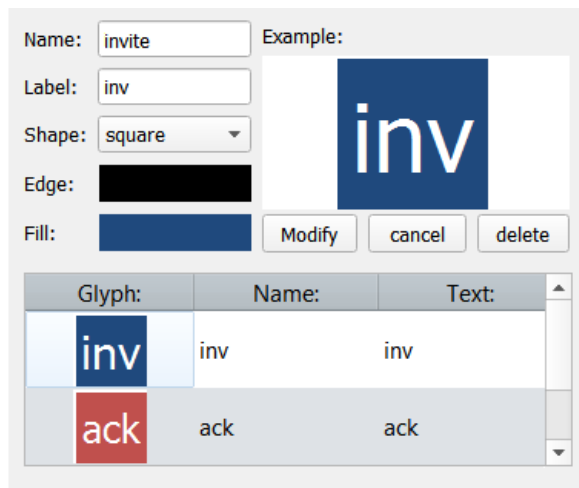


Figure 5.5: Editor interface for the construction of custom glyphs.

5

Users can specify a macroEventPredicate by selecting a glyph of choice. The “Wildcard” glyph corresponds to any label. Double-clicking a glyph in the query interface results in a textual interface (Figure 5.3). This enables users to specify a predicate over attributes and values in the data. For the right-hand side of a rule, users can design their own labels by choosing a particular shape, color and/or label (Figure 5.5). Earlier defined glyphs can be reused for the creation of new rules. In general, rules are used for three purposes (also depicted in Figure 5.3):

- ① filtering,
- ② highlighting, and
- ③ compression.

Filter rules are constructed by rewriting patterns to the empty macroEvent and are typically used for data reduction. Highlight rules enable users to visually emphasize events of interest for their investigation, whereas compression rules group collections of events to reduce variety or repetition.

5.5.3. Rule interaction

In Eventpad, multiple rules can be applied one after each other. Rules are shown in a list (Figure 5.4F) and applied from top to bottom. The ordering of the rules can be rearranged using drag & drop operations and rules can be toggled on and off to study their effect. Regex rules can match patterns of varying length, e.g., application of the rule $a.*b \rightarrow R$ to a string `abcb` can lead to either `Rcb` or just `R`. Analysts can determine for every rule whether longest or shortest matching should be applied using the rule view.

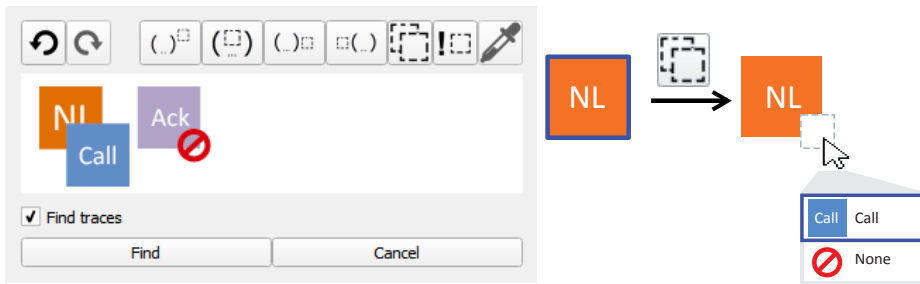


Figure 5.6: Search interface where users are only interested in sequences where call events also originate from the Netherlands. In addition, these call events should be followed by ack events only (i.e., without overlap).

A direct consequence of our extension to multivariate data is the possibility of having a macroEvent adhere to multiple rules at the same time (also referred to as *multi-matching*). This is for instance the case when two rules specify constraints over two different attributes. In order to make users aware of this, in Eventpad's sequence view we visualize this overlap by stacking the corresponding glyphs of a macroEvent on top of each other. An offset is used to ensure that previous matches are still visible to the user. Users can disable multi-matching to prevent glyphs from being stacked. In this situation, only the glyph of the last obtained macroEvent label is shown to the user.

To study the impact of a rule over the data set, an icicle plot is introduced visualizing the fraction of glyphs that are rewritten by the current rule set (Figure 5.4E). The icicle plot is constructed in the same order as the rules are applied. The order in which the rules occur in the rule view determines the ordering in which the glyphs are stacked.

Besides rules, users can select patterns via a search interface. Users can stack glyphs in the search interface to enforce a macroEvent to adhere to more than one rule at the same time. Inversely, the `exact` operator is used to ensure that macroEvents of interest only adhere to the specified visual representation. Figure 5.6 shows a search dialog where both operators are applied.

5.6. Pattern Aggregation

Although rules can significantly reduce the length and complexity of sequences, slight variations between sequences in real-world data are inevitable. In the next sections we describe how we use *clustering*, *alignment*, *sorting*, and *partitioning* operations to discover patterns between sequences through overview.

5.6.1. Structural sequence overlap

Depending on the type of rule it is possible for a glyph to represent more than one event. Users are informed about this by placing a red popup in the upper right corner of the glyph showing the number of events it contains. Since this popup does not alter the type of glyph, we decided to exclude this information during sorting and clustering.

Since all sequences typically do not fit on the screen, users can gain better insight in their event log by clustering sequences based on their glyph representation. This results in a stacked view only showing the unique sequences in the data set based on the currently defined set of rules. The frequency of a sequence is shown textually at the end of every sequence. Stacked sequences can be sorted by frequency to discover generic patterns or outliers. Users can also sort sequences according to various metrics:

- *Default* presents the sequences in the order of the input.
- *Alphabetical* sorts sequences by representing every sequence as a string. The string is obtained by concatenating the labels of glyphs together separated by delimiters.
- *Clustered* sorts sequences by applying single linkage agglomerative hierarchical clustering [93] on the unique event sequences based on Hamming distance.
- *Selected* sorts sequences by their number of currently selected events.

With hierarchical clustering sequences of similar length are positioned close to each other, whereas alphabetical sorts sequences based on their starting sequence. The Hamming distance $d(s_1, s_2)$ between two sequences s_1, s_2 is defined by their number of dissimilar glyphs:

$$\sum_{1 \leq i \leq \min(\#s_1, \#s_2)} \left(\begin{cases} 0, & s_1[i] = s_2[i] \\ 1, & \text{otherwise} \end{cases} \right) + |\#s_1 - \#s_2|,$$

where $s_1[i]$ represents the i -th macroEvent of s_1 and $\#s_1$ its corresponding length. Two macroEvents are assumed to be equal if they are rewritten by the same set of rules.

To gain better insight in characteristics of particular sequences, users are enabled to partition sequences according to a sequence attribute of choice. This enables users for instance to inspect event sequences based on their length, duration, or starting time. By default all event sequences are contained in a group called “All” (Figure 5.5A).

5.6.2. Sequential sequence overlap

The wide variety in event sequences sometimes makes it difficult to detect potential overlap between them. We believe however that discovering overlap is key to understanding how deviations between sequences may have developed. The alignment view (Figure 5.4B) enables analysts to generate an overview visualization of event sequences of interest by aggregating them in an icicle plot.

Current alignment methods [329] often focus on the alignment of sequences by the n -th occurrence of an event. In Eventpad, we assist analysts in finding multiple areas of interest through Multiple Sequence Alignment (i.e., MSA). Figure 5.7 shows the effect of MSA on the traditional icicle plot. MSA is a popular technique in the area of bioinformatics to discover patterns between (fragments of) DNA sequences. As DNA sequences typically show overlap in small fragments of the sequences, they often use MSA algorithms with local optimization.

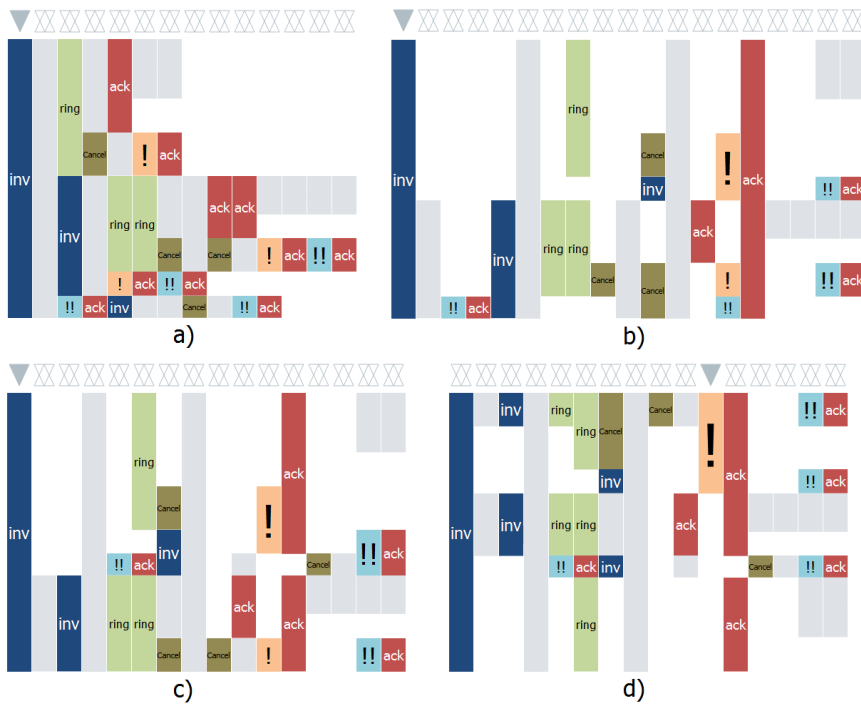


Figure 5.7: Effect of alignment to event sequences: a) No alignment. b) MSA layout with gap cost = 1 c) MSA layout with gap cost = 2 d) Effect of sorting on different part of the alignment.

Since we are interested in overlap and differences between entire sequences, we chose for an MSA algorithm focusing on global alignment.

Bose et al. showed the value of applying sequence alignment to gain better insight in event logs [31]. They also indicated the need for interaction to explore patterns in greater detail. In Eventpad analysts are enabled to apply the progressive global MSA algorithm by Bose et al. on event selections of interest to automatically find areas of overlap. Analysts can modify parameter settings such as gap cost to determine the amount of white spacing the MSA is allowed to introduce. Selecting a block in the alignment view results in the selection of its corresponding events in the sequence view. This enables users to select similar events across multiple sequences with a single click of a button.

5.7. Selections

Pattern aggregation enables users to perform high-level comparisons, but does not enable the inspection and comparison of multivariate data outside the scope of rules. In addition, analysts need to be enabled to focus on parts of the event sequences that are relevant for their investigations. For this we enable users to create selections of interest.



Figure 5.8: a) Application of INVITE (blue), ACK (bordeaux), and error messages (orange/blue). b) Messages in red represent external traffic.

5.7.1. Context

The context view (Figure 5.4D) enables users to save selected events of interest into a new context by assigning a name to them. The creation of a new context results in a new attribute separating the selected events from the non-selected. This attribute is added to the data and can be used for further analysis and drill down, enabling analysts to tag the data with more domain-specific information throughout exploration. To stay aware of the impact of a particular selection, the status bar is used to display the number of events and sequences that are currently selected in the active context.

Similar to Cappers et al. [46] contexts are saved in a tree structure where the hierarchy shows the ordering in which the contexts are created. Context c is a child of parent context d if and only if c was created when the analyst was exploring d . Contexts are used to focus on smaller subsets of the data. Right-clicking a context a while exploring b results in the selection of all events that a and b have in common.

5.7.2. Attributes

To enable the inspection of multivariate data in event selections of interest, an attribute view (Figure 5.4C) is introduced showing an overview of event attributes using scented widgets [333]. The histogram bins are interactive and can be used to select and deselect events with specific attribute values within the current context. Scented widgets for sequence attributes are introduced in a second tab. Selections can be enforced within specific value ranges of a widget by adjusting its scented span slider. For categorical attributes an exclamation mark is shown in front if the number of values is too large to visualize. To inspect selected values that are not visualized, histogram bins can be sorted by frequency, rarity, or whether they occur in the current selection. Since the number of attributes is typically larger than the number of widgets that fit on screen, we enable users to filter attributes by name using a textual interface.

5.7.3. Interaction

Event-oriented interaction is used to keep brushing and linking consistent and understandable over all the views. Users can modify selections of interest by selecting and deselecting visual elements across different views while holding the CTRL key. In case of the attribute

view, blue bars are used to show the fraction of selected events in every bin. Glyphs whose event collections are partially selected are marked with a blue dashed border, solid otherwise. Double-clicking a glyph in the sequence view results in the selection of its corresponding sequence. Right clicking on a selection enables users to store selections for further investigation, invert selections, or inspect the multivariate properties of the selected events in a tabular view.

5.8. Use cases

We demonstrate the exploration method on two real-world multivariate event sequence data sets. We show how tight coupling between multivariate and sequential analysis is achieved by starting exploration in the first example with the analysis of known sequential patterns in order to find anomalies and patterns in the multivariate data. In the exploration of the second data set we start with the analysis of multivariate data in order to discover patterns of interest in the sequences.

5.8.1. VoIP traffic

The proposed exploration method was designed in collaboration with a Dutch telecom company specialized in the provision of communication services over Internet using Voice over IP (VoIP).

Problem statement

For the establishment of a VoIP conversation handshaking signals such as invite (start call), acknowledge (accept call), cancel (interrupt call), and bye (end call) are transferred using a protocol called SIP [266]. Besides the type of signaling, these messages have additional information, such as status codes, source and destination phone numbers, (geo) domain information, user-agent etc. A conversation is uniquely defined by a `Call-Id`. The presence of illegal SIP sequences and/or invalid SIP messages in the traffic can cause SIP servers to go to an invalid state where conversations are no longer properly billed or secured, or could even lead to the disruption of the server [116].

A common attack model for hackers to abuse this state is to make money using Toll fraud [1]. In this model, hackers steal user credentials to hijack a company's VoIP phone, make many (long) phone calls to premium numbers they own in order to receive thousands of euros for the dialed numbers at the expense of the company. The detection of these Advanced Persistent Threats (APTs) [280] in general is difficult since APT and "normal" traffic look very similar.

The flexibility of the SIP protocol makes the distinction between valid and invalid phone conversations ill-defined. Depending on the vulnerabilities in new VoIP software updates, these definitions might even change over time. The main goal of the exploration is to find out whether their servers and customers properly send SIP messages conform the RFC standard [267]. Gaining insight in unexpected signaling can help analysts to understand whether they were caused due to bad server configurations or the presence of malicious users.

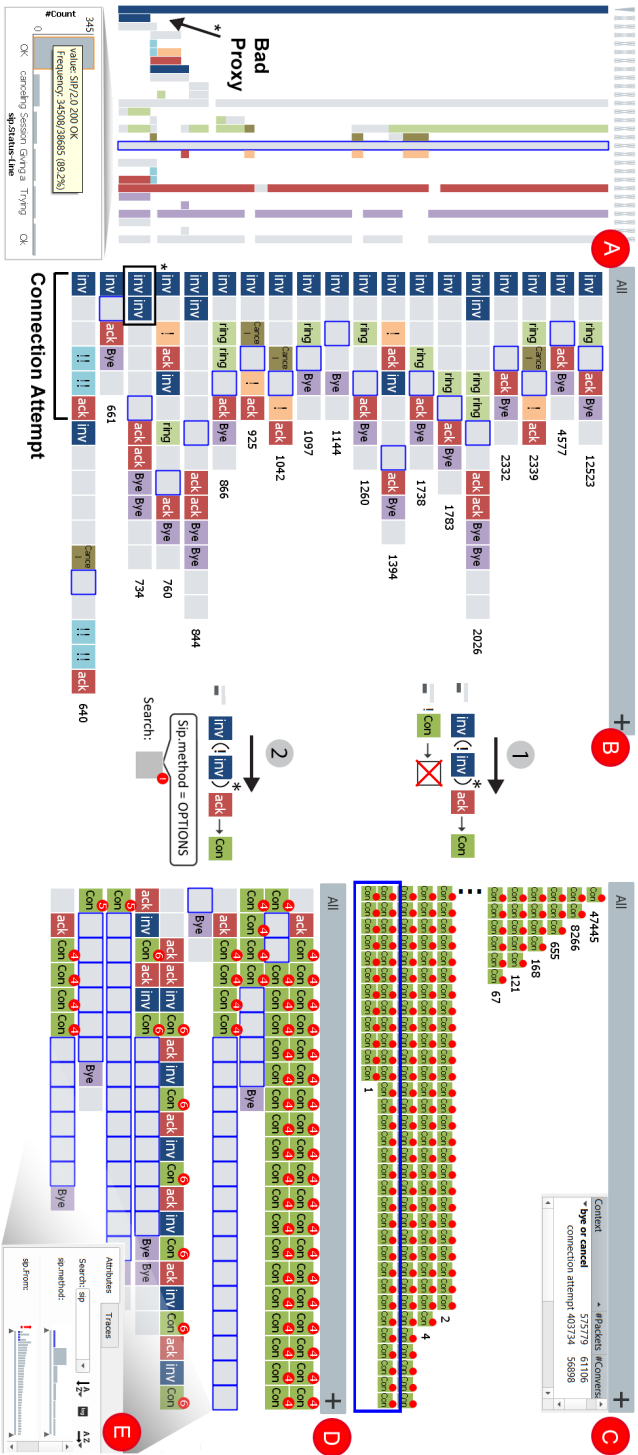


Figure 5.9: A) Investigation of multivariate data of aligned undefined elements. B) Most frequent VoIP patterns after preprocessing the data. C) Detection of many connection attempts within one conversation. D) Presence of ping traffic in conversations. E) Traffic originating from the same source.

Exploration

Together with four analysts we organized an interactive session where we used Eventpad to study their SIP traffic. In their daily routine the analysts use applications such as VoIPMonitor [282] and the protocol-analyzer Wireshark [63] to gain better insight in their phone traffic. They often use search tools such as `grep` and Elasticsearch in their investigations to find explanations for errors in their server logs. We initially analyzed 800,000 SIP messages consisting of approximately 181,000 conversations and 60 attributes. The traffic was obtained by recording 20 minutes of SIP signaling from one of the data centers with a load of approximately 3000 concurrent calls per second.

We started exploration with a *black listing* approach where analysts created rules to search for known undesired SIP conversations. For this they created a rule where glyphs whose event's status code represent SIP client or server failure are replaced with an orange "!" glyph. Blue "!!" glyphs are introduced for error codes that were sent from their own servers. The rule overview showed that only 2% of the events contained internal errors (Figure 5.8a). They investigated this by sorting the sequences alphabetically and creating three rules to visually distinguish between INVITE, ACK, and BYE messages.

The rule overview showed that only 35% of the data was covered by the new rules (Figure 5.8a). Inspecting the `sip.Method` widget showed that almost 60% of their traffic consisted of other SIP traffic involving `Option` messages to ping server information and `REGISTER` messages. In addition, they noticed a small percentage of `MESSAGE` events that are supposed to be deprecated in their platform because of known vulnerabilities [116].

To study the variety in the traffic, analysts decided to cluster the sequences and sort them by frequency. They were shocked to see that only 20% traffic was captured in the top 10 most frequent patterns (Figure 5.9B) as this indicates high variability and many deviations from expected standard behavior. Next, analysts filtered out incomplete conversations by searching for sequences starting with an invite request using the search interface. In addition, they decided to focus on their own traffic by excluding traffic from third party VoIP providers. This resulted in a new context and attribute called "Internal traffic".

After selecting the frequent patterns and aligning them using MSA, analysts saw that, despite the variety, overlap between sequences was high (Figure 5.4B). Analysts now noticed the presence of a proxy server in the middle of phone conversations (Figure 5.4B: two INV and two ACK messages nested). This made them realize that some phone calls were migrated to other data centers for load balancing. In cases where this proxy was not present, erroneous messages were generated twice in one sequence (highlighted in Figure 5.4B). In this sequence INV and ACK are not nested. Analysts selected the aligned "!!" events in the alignment view and inspected their multivariate data using a tabular view. Inspecting the `sip.From` header of the events revealed that most erroneous messages had an anonymous source phone number and an `invalid` domain (Figure 5.4C).

Based on previous observations, we organized a second session where we extended the analysis over a larger period in time. We excluded ping traffic, incomplete conversations due to fragmentation, and phone calls that were established outside their platform. In this session we incorporated two hours of traffic by simultaneously recording traffic from two data cen-

ters. Since Wireshark does not support any filtering mechanism at sequential level, we use the Eventpad engine for preprocessing the data set only considering conversations:

- that start with an `INVITE` message,
- where an invite should eventually be followed by a `BYE` or `CANCEL`, and
- only contains messages whose `From.host` and `To.host` are inside the company's domain.

In order to avoid the presence of duplicate messages due to call redirection of proxies, phone calls are identified by their triple `ip.src`, `ip.dst`, and `Call-Id`. The application of these filter rules resulted in the reduction from 40,000,000 events and approximately 4,000,000 conversations to the analysis of 1,300,000 events and approximately 80,000 conversations respectively.

After applying the rules, analysts created a new alignment of the most frequent patterns to see that in this selection the variety in the traffic was significantly reduced (Figure 5.9A). The sequential occurrence of `INVITE` messages in the Dutch traffic showed the presence of a bad proxy configuration where the target computer and proxy were the same (Figure 5.9A, *).

5

Partitioning the traffic by `geoip.src` showed that these bad proxies were located in Dutch traffic only (Figure 5.10). Furthermore, analysts noticed that most international phone calls to the Netherlands failed at their first connection attempt. Since it is possible for a phone conversation to have multiple connection attempts, analysts decided to extract these patterns by constructing a rule as specified in Figure 5.9B. Bad proxy settings were ignored using shortest matching and by enforcing that invites are not inside the attempts. After extracting the attempts and sorting them alphabetically, analysts noticed that most conversations in their network required at most two connection attempts in order to succeed (Figure 5.9C). Some phone conversations however required over 40 attempts (Figure 5.9D). Although the start time and duration of the conversation did not show anything suspicious, inspection of the event attributes shows the presence of `OPTIONS` messages inside a regular phone call. Although such a sequence is valid with respect to the RFC, in practice this is highly uncommon. After selecting all conversations with `OPTIONS` messages, sorting the attributes by relevance showed that all conversations originated from the same client (Figure 5.9E).

After the sessions, one of the analysts said “For us, the system is a business intelligence tool that can really help us in understanding what is actually happening in our platform.” Additional features such as integration with Wireshark and shortcut functionality to instantly remove selected patterns of interest was requested to speed up their analysis process.

5.8.2. Hospital records

To illustrate the effectiveness of the method in other domains, we also analyzed a real-world hospital data set provided by the BPIC11 contest [319] consisting of approximately 1000 sequences, 134,000 events, and approximately 130 attributes. Apart from anonymization, the hospital log stores for every patient of a Gynaecology department when certain activities took place along with additional attributes such as which group performed the activity, the

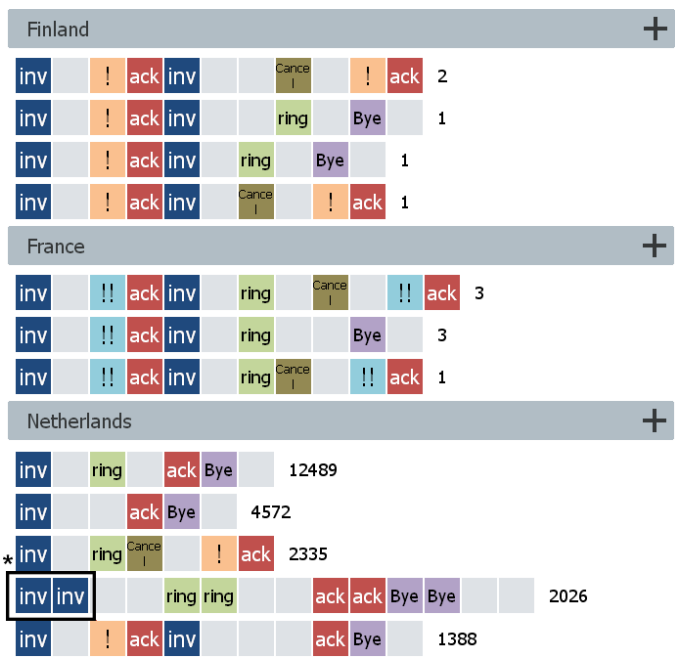


Figure 5.10: Partitioning by `geoip.src` reveals that external calls contain errors and the presence of bad proxy configurations (*) in Dutch traffic.

age of the patient, the activity’s level of emergency etc. The hospital log contains patient treatments where urgent activities are performed. In this use case we want to know when the hospital decides to make certain activities urgent. We try to find an explanation for these activities by testing whether urgent patient treatments share particular events.

We start exploration by first creating a rule where all urgent events are marked in red (indicated in Figure 5.11B). We obtain all urgent sequences by searching for sequences with at least one red glyph and store them in a new context “urgent” (corresponding to 256 sequences). After selecting the new context, we inspect the `org_group` widget to see that all urgent events occur in the General Lab. To discover whether urgent events happened before or after certain treatments in other departments, we select events based on their `org_group` using the attribute view and inspect the number of cases that are involved via the status bar. This showed that in approximately half of the cases, the Radiotherapy group was involved in the treatment process.

Since radiotherapy is exceptional and only used for the treatment of cancer, we want to know whether the urgent activities happened before, after, or during this treatment. To explain the location of these events, we first have to understand the general workflow inside the department. We extract the radiotherapy events from the event log using the search interface and store results in a new selection “radio”. We investigate the different activities using the `event:name` attribute widget. The number of different activities in the department is too large to visualize individually, but we can see that the most frequent activities can



Figure 5.11: A) Investigation of anomalous alignments using selections and attribute inspection. B) Rewrite rules to study workflow patterns in the Radiotherapy department: 1. Group successive consultancy (blue) events in one glyph 2. Filter by radiotherapy (yellow) 3. Disable radiotherapy encoding. C) MSA results with gap cost 2 after incrementally adding knowledge into the data.

be grouped in four categories: consultancy (e.g., primary, secondary, and consultancy by phone), teletherapy (external radiation treatment), brachytherapy (internal radiation treatment), and payment administration. We define these categories by defining four rules where we search for events with keywords “consult”, “teletherapy”, “brachytherapy”, and “rate” in their activity name respectively. Since consultancy may require multiple sessions in a row that does not add value to the investigation, we define an additional rule where subsequent consultancy events are represented by a single blue glyph. Figure 5.11B shows the events of interest after applying the rules. Figure 5.11C shows the result of applying MSA to the glyph sequences before and after defining every rule one by one.

Applying MSA with gap cost of 2 shows that urgent radiotherapy treatments in general first start with consultancy (blue), basic treatment (gray), teletherapy (green) after which there

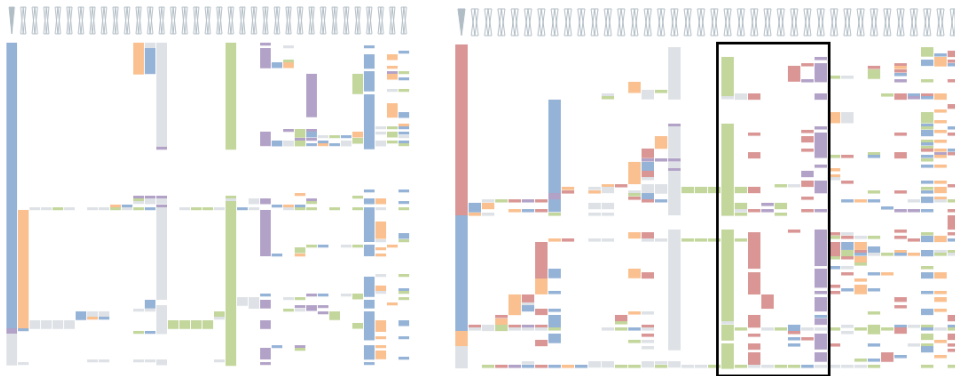


Figure 5.12: General workflow radiotherapy department without (left) and with (right) urgent events in red. Urgent events between teletherapy (green) and brachytherapy (purple) mainly test for kalium, leukocytes and trombocytes in the patient's blood. Urgent events before and after therapy focus on calcium, glucose, natrium etc. Orange and blue events represent payment and consultancy respectively.

is a concluding consultancy. Figure 5.12 shows the radiotherapy alignment along with the emergency activities indicated in red. Here we can see that in approximately half of the cases emergency events occurred before radiotherapy was started. We can also see that urgent events often occur between teletherapy (green) and brachytherapy (purple) sessions (Figure 5.12, black box). Inspection of these events shows that they analyze trombocytes and leukocytes in the patient's blood to study the effect of the radiation. In non-urgent cases we discovered that this type of blood research is only performed during annual consultancies. This shows that the hospital assigns a higher priority to blood results of patients during radiation treatment. Patients who still had urgent activities after brachytherapy were all diagnosed with either gynecological tumors or cervix uteri cancer (both highly uncommon in the data set).

Besides the general workflow, we discovered some anomalies that were unexpected:

- Some treatments involving brachytherapy did not receive consultancy in the Radiotherapy department (Figure 5.11A-3). Inspecting the sequences of interest outside Radiotherapy reveals that consultancy happened in a different department (Figure 5.11B).
- Some treatments show an increased number of undefined and teletherapy sessions after each other (Figure 5.11A-2). Closer inspection of the sequences in a tabular view reveals that treatments involve a rare activity named “simulator” (Figure 5.11A-1).
- In most cases brachytherapy is only performed after a teletherapy session (Figure 5.11A-3). However in 3% percent of the sequences this did not hold (Figure 5.11, red circles). In these situations, all patients were diagnosed with *Malignancy endometrium* whose treatment apparently started all on the same day.

5.9. Discussion and Limitations

The use cases in Section 5.8 show how combined exploration of attributes and sequential analysis is achieved using rule-based event rewriting, sequence aggregation, and selections of interest as central elements. Each of these concepts has its own advantages and limitations for the analysis of events, we show that when combined they amplify each other. Tight linking between the concepts provides analysts a minimalistic yet expressive visual query mechanism to interactively select sequences of interest based on their sequential attributes, event patterns, and multivariate data associated with these events.

The definition of an anomalous sequence in general is ill-defined and requires domain knowledge and multiple iterations to construct properly. The ability to visually encode parts of the data based on rules enables analysts to incrementally label their data and define their notion of what a good or bad sequence should look like. By sorting and clustering events based on the specified visual encodings analysts are able to study the coverage of their rules and discover new patterns of interest.

5

Furthermore, the ability to evaluate rules in an offline setting also makes the tool suitable for data cleansing applications [207], where analysts can use the rules to only obtain the parts of the data (sequences) that are relevant for their investigation.

As with every technique, there are limitations. Although the application and construction of a rule in general is easy and intuitive to understand, the resulting rewriting can become complex when evaluating a large number of rules. During the interactive sessions with analysts we noticed that they needed about 10 active rules in order to answer their questions. The application of too many rules in parallel however can clutter the sequence and alignment view. Analysts can replace multiple rules by a new one by combining constraints over multiple attributes in one rule, but this only solves the problem partly. Although regular expressions in general are powerful to specify patterns, they have difficulties in capturing non-deterministic behavior. Especially for the analysis of systems involving parallel communication, the specification of patterns using regex can become cumbersome.

The effectiveness of clustering and alignment methods depends on the amount of information that is incorporated in the definition of the rules. In case of the hospital data we saw for instance that naively applying MSA without any proper rules resulted in no insights (underfitting) whereas creating rules for all possible (combinations of) values is tedious and results into noisy alignments (overfitting). Although it is possible for analysts to simplify sequences using rules and filtering techniques, domain knowledge about the underlying data is essential to obtain desired results. Especially when analyzing long sequences, coping with the variety in sequences and multivariate data can become difficult without a priori knowledge and expectations.

The approach suffers from a “cold-start problem” in the sense that users must already be aware of the information to be queried [212]. Although selections and scented widgets can help analysts in finding characteristics in parts of the data that are not captured by rules, this can be time-consuming without (automated) guidelines.

The performance of progressive MSA and single linkage hierarchical clustering in general are $\mathcal{O}(n^2m^2)$ and $\mathcal{O}(n^2m)$ respectively, where n represents the number of sequences and m the length of the sequences of interest. In practice we noticed that the application of the techniques to only unique patterns in the data set and selections of interest makes the analysis interactive for hundreds of sequences. However, interactivity can be affected when naively applying these techniques over larger collections of long and unique sequences.

With respect to the visualization, the presented exploration approach focuses on the sequential occurrence of events rather than the time between them. In our problem statement absolute time does not play a major role. For applications where the time between events is relevant for the analysis, the interface has to be adapted. Finally, if the number of attributes in events and sequences is large, interaction with attributes is limited to the number of visible scented widgets. Although sorting, filtering, and scrolling helps at finding attributes of interest, specifying queries involving many attributes becomes time-consuming.

5.10. Conclusion and Future Work

5

We presented a novel approach for analysts to explore multivariate event sequence data by combining attribute and sequential analysis into one unified system. The ability to interactively encode event logs by coalescing event sequences according to rules enables analysts to incorporate their knowledge to the data and test whether this matches their expectations. The combination of attribute-based scented widgets and pattern aggregations enables analysts to discover new attributes of interest and refine rules based on their new findings while staying aware of high-level patterns across different levels of abstractions. We have shown the effectiveness of the approach on real-world data sets through interactive sessions with external companies and elaborate examples. Since the methodology makes no underlying assumptions on sequential data, it is general and flexible enough to be used in other domains.

For future work it is interesting to see how we can extend the visualization paradigm to support more complex regex operators such as *capture groups* and *back referencing* [111]. Also, an extension of the glyph design interface to enable further parametrization of glyphs can help to study correlations between two or more attributes.

The system currently focuses on defining new attributes of interest at the level of an event. Applications where multivariate data is difficult to obtain (e.g., flow-based network traffic analysis [197]) however often focus on sequential properties in their analysis. Defining proper interaction schemes to support the creation of attributes at sequential level however is nontrivial, since they apply to a different level of abstraction.

Appendix

Regular expressions for multivariate data

Let E be the set of all possible microEvents and let A and V represent the set of all possible event attributes and values in the data set respectively. Let $Exp \in B$ be any Boolean expression over attributes and values in A and V . Furthermore, let s be a sequence of macroEvents, where $[]$ represents the empty sequence and $s = b \triangleright s'$ indicates that b is the first macroEvent in s . Let $s = s_1 \uplus s_2$ indicate that s is the concatenation of subsequences s_1 and s_2 . For the sake of simplicity, we assume that every macroEvent belongs to exactly one sequence. Finally, let $SAT(e, Exp)$ return whether macroEvent e satisfies Boolean expression Exp . We can now formulate the satisfiability of a regular expression R in a sequence s as the definition of a Boolean function $RSAT(s, R)$:

5

$$RSAT([], Exp) = false$$

$$RSAT([], \epsilon) = true$$

$$RSAT(b \triangleright s', R) = \begin{cases} R = Exp \Rightarrow SAT(b, Exp) \wedge s' = [] \\ R = R1 + R2 \Rightarrow RSAT(s, R1) \vee RSAT(s, R2) \\ R = R1 . R2 \Rightarrow \exists_{s_1, s_2} RSAT(s_1, R1) \wedge RSAT(s_2, R2) \wedge s = s_1 \uplus s_2 , \\ R = (R1)^* \Rightarrow RSAT(s, \epsilon) \vee RSAT(s, R1 . (R1)^*) \\ R = \epsilon \Rightarrow false \end{cases}$$

where operators $+$, $.$, and $*$ represent choice, sequential composition, and iteration (0 or more times) respectively. ϵ represents the empty regular expression.

Rule evaluation

In traditional regular expressions, a character in a regular expression and input string match each other if they are equal. However, in regular expressions for multivariate data, we reason about macroEvents and macroEventPredicates.

Let g be a macroEvent and P a macroEventPredicate. The evaluation of a regular expression with predicate logic can be directly mapped to the evaluation of a traditional regular expression by defining when macroEvents in g satisfy P . In case of *minimal* matching, at least one microEvent in g should satisfy P .

To formalize this rule evaluation, let $\mathcal{L} \cup \{Wildcard\}$ be the set of all possible labels, and let GR be a regular expression consisting one macroEventPredicate P and possibly one or more operators. We say that a macroEvent g satisfies expression GR if and only if $GRSAT(g, GR)$ holds:

$$GRSAT(g, GR) = \begin{cases} GR = !GR' \Rightarrow \neg GRSAT(g, GR') \\ GR = \exists \text{ exact } P \Rightarrow GRSAT(g, P, \exists, true) \\ GR = \forall \text{ exact } P \Rightarrow GRSAT(g, P, \forall, true) , \\ GR = \exists P \Rightarrow GRSAT(g, P, \exists, false) \\ GR = \forall P \Rightarrow GRSAT(g, P, \forall, false) \end{cases}$$

where $!$ represents negation and $GRSAT$ evaluates whether g matches P as specified in the expression. The operators \exists and \forall specify whether macroEventPredicates should be evaluated according to minimal versus maximal matching. The `exact` operator specifies whether macroEvents are allowed to have additional labels that are not specified in the macroEvent-Predicate. In order to define $GRSAT$, let:

- $Labels(P)$ be the Boolean expression of P over labels in $\mathcal{L} \cup \{Wildcard\}$, and
- $Expr(P)$ be the Boolean expression of P over attributes and values in A and V .

The function $GRSAT(g = \langle L, ES \rangle, P, op, exact)$ is defined as follows:

$$GRSAT(g, P, op, exact) = \begin{cases} exact \Rightarrow Labels(P) = L \vee (Labels(P) = \{Wildcard\}) \\ \neg exact \Rightarrow Labels(P) \subseteq L \vee (Labels(P) = \{Wildcard\}) \end{cases} \wedge \begin{cases} op = \exists \Rightarrow \exists_{me \in ES} SAT(me, Expr(P)) \\ op = \forall \Rightarrow \forall_{me \in ES} SAT(me, Expr(P)) \end{cases}$$

where $\langle L, ES \rangle$ refer to the set of labels and microEvents of macroEvent g .

The wildcard label matches any label. The evaluation scheme does not require labels to be represented as glyphs, since all operators in $GEXPR$ can be directly translated to the evaluation of sets of microEvents. This enables analysts to pre- or post process event data without the use of an intermediate visual representation.



Hypothesis Testing & Generation in Wildlife traffic

6

This Chapter is based on [43]:

B.C.M. Cappers. Exploring Lekagul Sensor Data using Rules Aggregations and Selections

In Proceedings of the IEEE Visual Analytics Science and Technology Challenge 2017

(Visual Analytics and Science Challenge 2017 Award “Elegant Tool for Hypothesis Testing and Generation”)

6.1. Hypothesis testing in Lekagul sensor Traffic

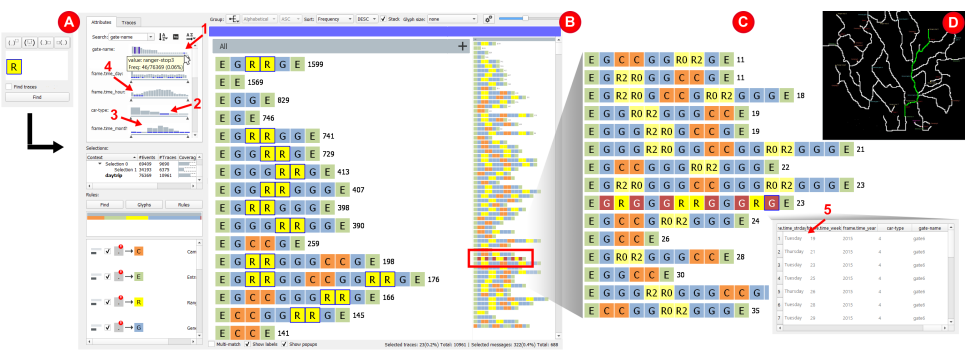


Figure 6.1: A) Search for rangerstops B) Deviating pattern in overview C) Inspection shows the presence of a 4Axle trucks in range routes. D) Graphical representation of taken route.

Chapter 4 showed how we can use visualization to determine the relevance of contextual anomalies by inspecting them from different angles. In Chapter 5 we focused on the visualization of collective anomalies by inspecting patterns within sequences, between sequences, and inside event properties. In this chapter we illustrate how we can use the Eventpad prototype to explore both contextual and collective anomalies in vehicle traffic data from the Visual Analytics Science and Technology Challenge 2017 [67, 68].

6.2. Introduction

The Boonsong Lekagul is a nature preserve that is used by local residents and tourists for day-trips, overnight camping, or as a passage to get to the other side. The gates inside the habitat are monitored to detect suspicious activity in the wildlife or vehicle traffic movement. An event is generated each time a vehicle drives through one of the gates.

Researchers of the preserve at some point discovered that the Rose-crested Blue Pipit bird species is no longer nesting and slowly dying out. They want to investigate if there is a link between this event and the traffic movement. Speeding and driving overnight for instance can scare away the wildlife. The goal of the VAST Mini Challenge 1 was about answering the following question:

What are the top 3 patterns that could be most impactful to bird life in the nature preserve?

In order to study anomalous behavior in the multivariate event sequences, we first need to know what normal behavior looks like. To this end, the challenge also required to answer the following subquestions:

Q1: Describe up to six *daily* regular patterns of vehicle driving through and within the park.

Q2: Describe up to six vehicle traveling patterns that occur *over multiple days* (e.g., weekly, annual, seasonal patterns).

Q3: Describe up to six patterns of activity (either single day or multiple days) that are *unusual* or difficult to explain.

For every pattern we had to specify the type of vehicles that were involved, where they went, and when the pattern happened. In addition, we had to come up with hypotheses that could explain the presence of the discovered patterns.

This chapter is structured as follows. First an overview of the data set is provided. Section 6.4 presents a visual analytics approach to solve the challenge through hypothesis testing. In Sections 6.5, 6.6, and 6.7 we present the patterns that were discovered using the approach. Finally, the challenge question is answered and summarized in Section 6.8.

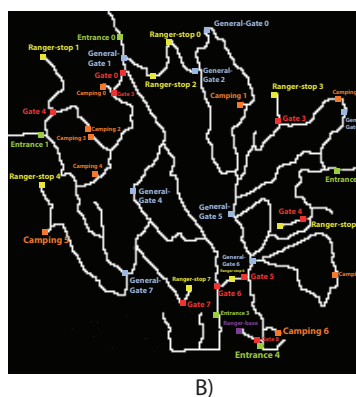
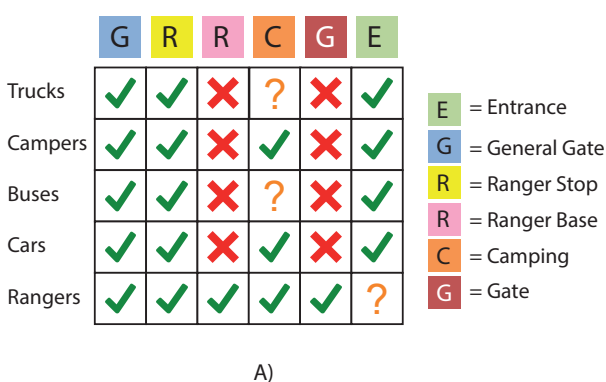


Figure 6.2: A) Access-control matrix showing the type of vehicles that are allowed in different parts of the park. B) A geographical map of the Boonsong Lekagul Natural Preserve.

6.3. Data

The Lekagul Natural Preserve records sensor events for every vehicle driving through gates inside the habitat. Besides a timestamp and car-id, events store additional data such the type of vehicle (e.g., car, truck, bus) and the gate name.

The data set consists of approximately 170,000 events and 18,000 different vehicles describing one year of traffic movement. Besides the historical data we also received a geographical map of the area of the area along with a description of the different types of gates inside the park (Figure 6.2b). The map is a bitmap file of 200 by 200 pixels corresponding to an area of 12 miles by 12 miles.

The data set identifies different types of vehicles such as buses, trucks (either transport trucks or campers), cars, motorcycles, and ranger vehicles. Ranger vehicles are special purpose vehicles that are used by the Lekagul personnel to drive inside the preserve.

The wildlife preserve consists of 47 gates belonging to one of the categories:

- *Entrances*: All vehicles drive through an entrance when entering or leaving the preserve.
- *General gates*: General gates provide valuable sensor data. All vehicles are allowed to drive through these gates.
- *Gates*: Only ranger vehicles are allowed to use these gates.
- *Ranger-stops*: These sensors represent construction areas for the rangers.
- *Campings*: Visitors pass these gates when entering the campground.

Finally, there are regulations inside the park:

- The maximum speed limit is 25 mph.
- Traffic either drives through the preserve, stay as day campers, or stay as extended campers.

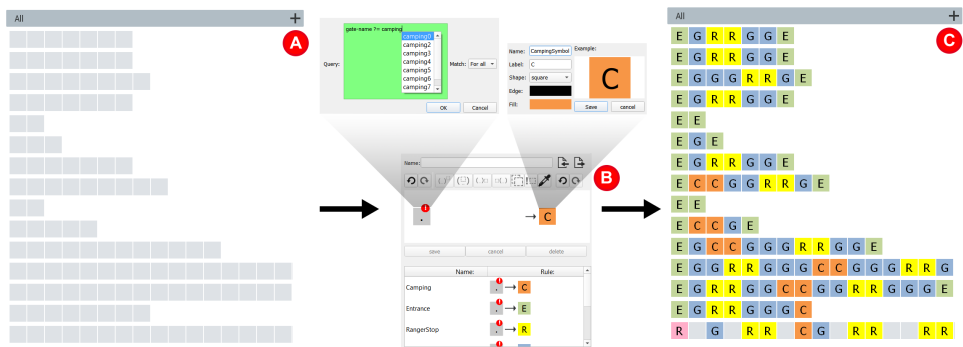


Figure 6.3: A) Event sequences are grouped by `car_id` and visualized as block sequences. B) Construction of rules using regular expressions and logic. C) Result after rule application.

6.4. Exploration

In order to answer the main research questions of the challenge, we need to understand what travel patterns are considered normal. According to the challenge description, we know that certain vehicle types have restricted access in the preserve. Figure 6.2a shows an access-control matrix describing the gates every car type is allowed to use. However, the description is not clear about buses visiting campings or whether ranger vehicles are allowed to leave the preserve. Combined with the speed limit restriction in the park, this raises questions such as:

- Are there vehicles speeding on certain roads?
- Is it possible for trucks to visit campings or ranger-stops?
- Are ranger vehicles always staying inside the preserve?
- Is it normal for visitors to stay for more than a week in the preserve? And do they eventually leave?

- How many times do vehicles visit the preserve in a year?

To answer these questions we used the Eventpad system for quick verification of these questions using rules, aggregations, and selections. In Eventpad we can map every sensor event to a glyph (Figure 6.3A) and group events into sequences by an attribute of choice. This enables users to inspect data from different points of view. Furthermore, the system can also use the event timestamp information to derive additional metadata such as the hour, day, week, and year of the event's occurrence. For a demonstration of the system in practice, we refer to the supplementary video¹.

In Eventpad we can search and color glyphs based on event attributes of interest using regular expressions and predicate logic (Figure 6.3C). We start the analysis by creating 5 rules coloring all camping events orange, entrances green, general-gates blues, rangerstops yellow, and rangerbase events pink. All time and date formats are stated in the format “day-month-year hours:minutes” (24 hours notation).

6.5. Q1: Daily patterns

For the inspection of frequent daily patterns in the data, we group the events by `car-id` and `date` such that a sequence represents the travel pattern of a vehicle per day. We study frequent patterns in the data by clustering the sequences based on their visual representation and sorting them by frequency. The result is shown in Figure 6.4A.



Figure 6.4: A) Vehicle patterns as shown as sequences of glyphs. B) Scented widgets of (derived) attributes. C) The Rule view shows an overview of the applied rules. D) Eventpad enables the discovery of patterns between sequences using data alignment.

¹<https://www.youtube.com/watch?v=IBgJ3R9cAvQ>













Pattern	Frequency	Target Group	Regex Pattern
1.From a camping to an exit	≈26%	≈ 50% 2Axle cars ≈ 30% 2Axle trucks ≈ 20% 3Axle trucks	 () * 
2.From an entrance to a camping	≈25%	≈ 50% 2Axle cars ≈ 30% 2Axle trucks ≈ 20% 3Axle trucks	 () * 
3.Day trips through the preserve	≈ 43%	≈ 40% 2Axle cars ≈ 23% 2Axle, 3Axle trucks ≈ 15% 4Axle trucks, 2Axle, 3Axle buses	 () * 
4.Ranger roundtrips	≈ 4%	2P traffic only	 () * 

Table 6.1: Frequent patterns in 25523 sequences. Most of the patterns were done by 2Axle cars. The patterns are matched over entire sequences only.

Applying Multiple Sequence Alignment [31] on the most frequent sequences enables us to identify four main patterns, namely vehicles entering (Figure 6.4B-1), leaving (Figure 6.4B-2), driving through the preserve (Figure 6.4B-3), and ranger traffic (Figure 6.4B-4). We can count the frequency of these patterns using regular expressions. Table 6.1 shows the frequency of the four patterns for different types of vehicles. Sequences that do not match patterns in Table 6.1 are vehicles driving overnight. These are discussed in Section 6.6. Next, we explore the daily patterns in greater detail.

Daily patterns entering and leaving campings

We can search for all sequences visiting campings by searching for the orange blocks using Eventpad's find functionality. After searching, the Attribute view in Figure 6.5A-2 shows that vehicles don't enter the campings between 02:00 and 04:00. Maybe vehicles are not allowed to enter campings in the middle of the night (e.g., curfew). Furthermore, campings are only visited by 2Axle cars, 2Axle trucks, 3Axle trucks, and ranger vehicles corresponding to the Matrix 6.2a in Section 6.4. This is also illustrated in Figure 6.5A-1.

Extraction of the camping events (Figure 6.5B) using the search interface and selecting the trucks in the Attribute view shows that trucks visit at most three campings per day. The dashed lines around the selected sequences show that there are also other vehicle types with these patterns. Coloring the blocks using a more detailed rule set shows that `Camping1` is least visited (Figure 6.5C-3). Figure 6.5C-4 also shows that the camping is not visited in December, March, and April.

Daily patterns in day trips

We can inspect the day trip traffic by looking for all sequences whose first and last events are entrance events. Since we did not see any 4Axle trucks and buses driving to campings, we decided to inspect these vehicles in greater detail by creating a filter in the Context view (Figure 6.6A).

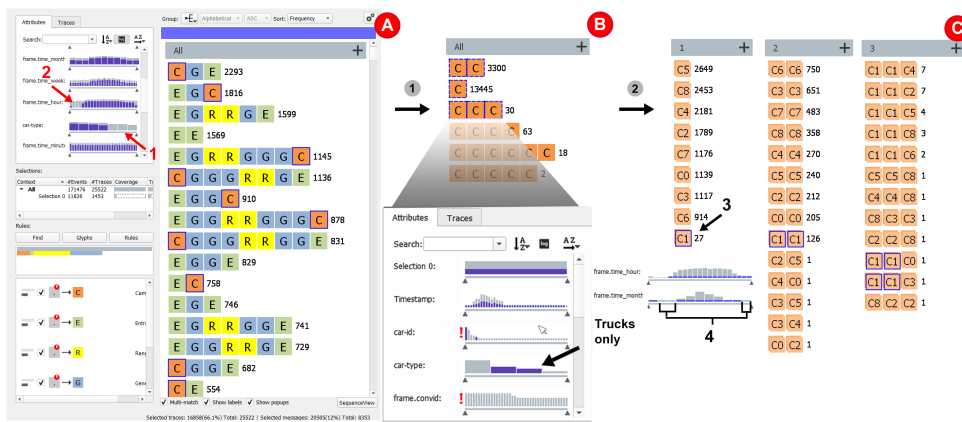


Figure 6.5: A) Searching for camping events in vehicle patterns entering and exiting campings. B) Extraction of the camping events shows that buses and truck visit at most three campings per day. C) Grouping the sequences by length shows that `Camping1` is least visited.

The sequence clustering shows 450 cases of trucks and buses driving through the preserve without visiting other gates. After selecting the bus sequences only, shown as selected sequences in Figure 6.6A, we see that these vehicles only visit the preserve once per day (i.e., there are at most two entrance event per sequence).

After loading a more detailed rule set, we can see the number of times sequences start and end at different entrances (Figure 6.6B). The alignment in Figure 6.6C shows that apart from one sequence the data is nicely structured. The long sequence represents the unauthorized access of a 4Axle truck which is discussed in Section 6.7.



Figure 6.6: A) Inspection of 4Axle truck and bus patterns. The travel patterns by buses are highlighted in blue. B) Application of detailed ruleset. C) The alignment of bus and truck traffic shows a suspicious long sequence (indicated by the black box).

Traffic speed

We can analyze the traffic speed of vehicles that do not stay in campings by dividing the duration of vehicles routes by the distance they have traveled on the map. The map shows that the fastest route, involving the least number of gates, from north to south is from Entrance0 to Entrance3 (Figure 6.7C). Similarly, the fastest route from east to west is from Entrance2 to Entrance4.

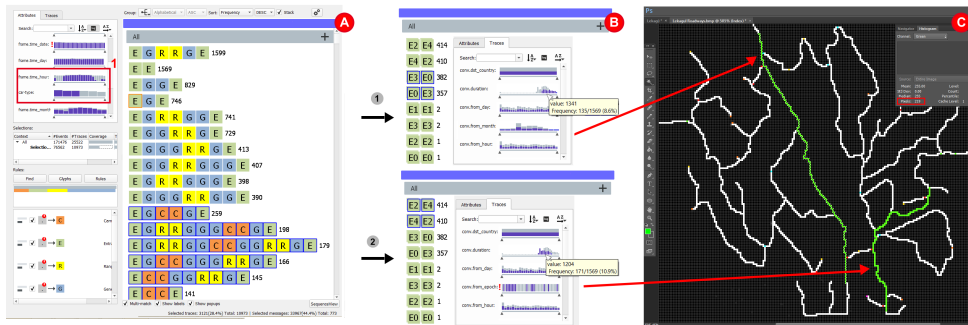


Figure 6.7: A) Sequences from and to an entrance. B) The travel time between routes can be analyzed by inspecting the sequential properties of the routes in the Attribute view. C) The distance between two gates is calculated using the color histograms in Adobe Photoshop.

6

Figure 6.7A shows an overview of all vehicle patterns starting and ending with entrance events. Selecting the sequences with camping events shows that 28% of the sequences visit campings. These vehicles correspond to 2Axle cars, 2Axle trucks, and 3Axle trucks. Figure 6.7A-1 shows that these vehicle do not visit campings between:

- 02:00-04:00 and
- 20:00-21:00.

Possible explanations are that people are having dinner between 20:00-21:00, campings are closed after 01:00, or people are sleeping between 02:00-04:00.

Most of the other traffic travels directly from entrance to exit. After loading a more detailed rule set, we can see that most of the direct routes correspond to the main roads (E0, E3) and (E4, E2) (Figure 6.7B). Selection of the routes between E0 and E3 shows in the Attribute view that the route on average takes 1341 seconds to drive (Figure 6.7B). Using Photoshop histograms we know that the travel distance is 219 pixels (Figure 6.7C). Knowing that 200 pixels on the map correspond to 12 miles, we can calculate the average speed as follows:

$$\frac{\text{distance}}{\text{time}} = \frac{219 \text{ pixels} \times (12 \text{ miles}/200 \text{ pixels})}{1341 \text{ seconds} \times (1 \text{ hour}/3600 \text{ seconds})} \approx 35 \text{ mph}$$

which is in violation with the 25 mph speed limit. Vehicles that drive between E2 and E4 do not exceed the 25 mph limit.

Daily patterns in ranger shifts

In contrast to regular traffic, ranger vehicles always start and end in the Rangerbase (Figure 6.8A). In addition, Figure 6.8C shows that they do not travel between 04:00-05:00 in the morning. The absence of entrance events confirms that ranger vehicles are not leaving the preserve.

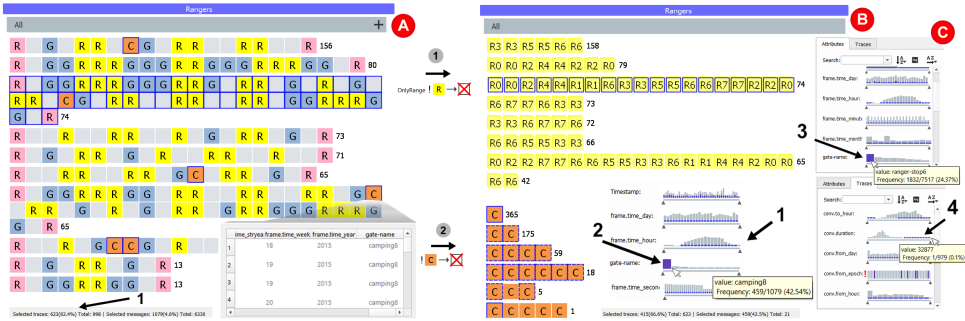


Figure 6.8: A) Frequent ranger shifts. B) Extraction of rangerstops. C) Properties of longest shift.

Figure 6.8A-1 shows that 62% of the ranger shifts visit campings between 06:00-22:00. In almost half of these shifts, rangers drive to Camping8. We can observe this by filtering the traffic by camping events and checking the frequencies of all campings in Eventpad's Attribute view (also shown in Figure 6.8B-2). After filtering the sequences by ranger stops and clustering the sequences using a detailed rule set, we can see that Rangerstop6 is most frequent in the yellow sequences. This is probably the fastest way to get from the west side of the park back to the base (Figure 6.8C-3). The longest ranger shift is approximately 9 hours visiting almost all stops (Figure 6.8C-4).

Car-type	Duration distribution (log)	Normal	Outliers
2Axle cars		Between 0.5 hour and 37 days	2 minutes, 350 days
2Axle trucks		Between 0.5 hour and 20 days	4 minutes, 107 days
3Axle trucks		Between 0.5 hour and 10 days	4 minutes, 23 days
2P vehicles, 4Axle trucks, and buses		Between 0.5 and 10 hours	12 hours

Table 6.2: Travel duration of vehicles in the preserve. The outliers on the right in the histograms correspond to vehicles either staying very long inside the park or visiting the preserve multiple times per year.

6.6. Q2: Periodical patterns

To study weekly, monthly, and seasonal patterns in travel history, events are grouped by `car-id`, such that every sequence shows the travel activity of a car throughout the year. This enables us to find patterns inside the sequences and discover signs of repetitive behavior.

We can inspect the start and end time of every sequence in the Attribute view. Table 6.2 shows the travel time distribution of vehicles in the data set. In the rightmost column we can see that there are 2Axle trucks and cars whose time difference between the first time entering and last time leaving is more than 100 days. Since it is possible for a vehicle to enter and exit the preserve multiple times in such timespan, we have to check whether this is indeed the case.

Number of visits

We can count the number of visits inside the sequences by creating a rewrite rule that replaces every subsequence of entrance and exit events in a purple block. To ensure that the rule does this in a non-greedy way, we state that in between an entrance and exit event another entrance event is not allowed (Figure 6.9B).

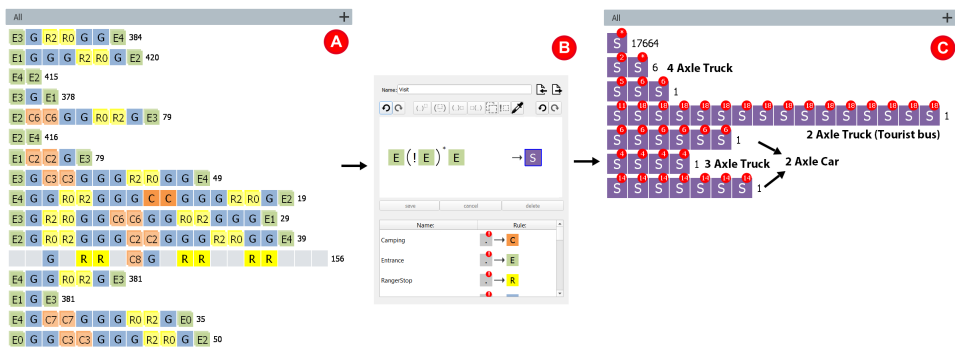


Figure 6.9: A) Event sequences are grouped by `car-id` only. B) We can count the number of visits by rewriting subsequent entrance events to purple blocks. C) Result after applying the rules. The purple sequences show that there are vehicles visiting the preserve multiple times in one year.

Filtering the purple blocks shows that most vehicles enter and leave the preserve at most two times (Figure 6.9C). The long sequences belong to different vehicle types. In the next paragraphs we inspect the three longest sequences in detail to find potential explanations.

Tourist bus activity?

The longest purple sequence corresponds to a 2Axle truck that has been active in the park for 107 days. We can inspect this sequence in greater detail by disabling the purple block rewrite rule and loading in a more detailed rule set. Figure 6.10A shows that the truck only drives between `Entrance4` and `Camping4` between 23:00-00:00 and 14:00-15:00 (Figure 6.10A-1). Since the vehicle only drives this path during high-season (July–October, Figure 6.10A-2) covering a large part of the preserve, we believe it could be a tourist bus or some taxi service. For unknown reason, the vehicle only travels on Sundays, Mondays, and Fridays (Figure 6.10A-3).

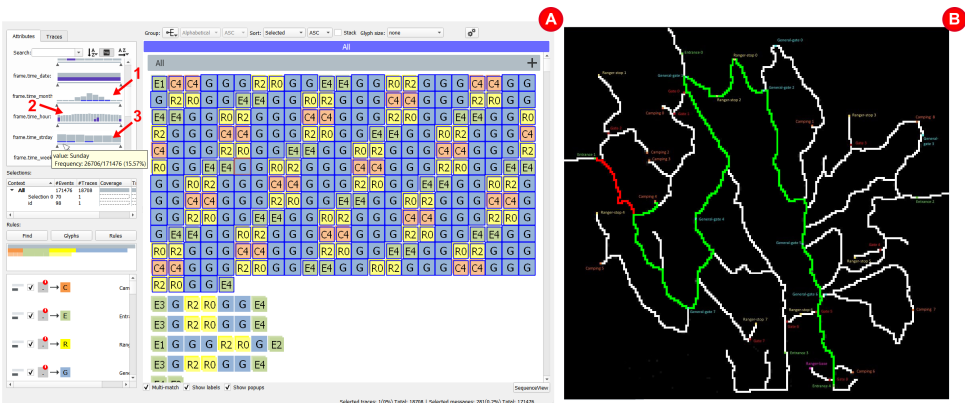


Figure 6.10: A) Weekly pattern of tourist bus in preserve. B) Geographical representation of the vehicle route. Route in red is the place where the bus entered the preserve before it started driving between Entrance4 and Camping4.

Stationary camper?

The second longest purple block sequence corresponds to a sequence of 37 days (Figure 6.11A-1) in which one 2Axle car (Figure 6.11A-2) travels on Sundays and Fridays (Figure 6.11A-3 from Entrance0 to Camping6 and vice versa.

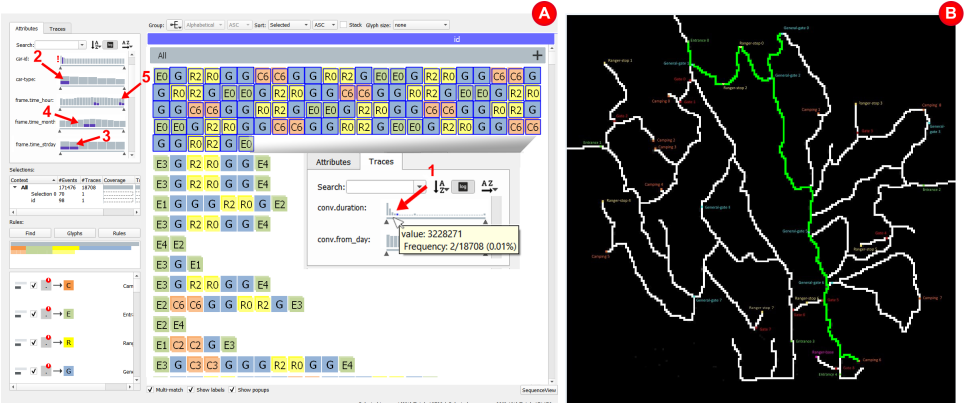


Figure 6.11: A) Weekly pattern vehicle visiting camping. B) Geographical representation of the travel pattern.

The vehicle only travels in the period 24-06-2015 until 30-07-2015 (Figure 6.11A-4) between 13:00-14:00 and 22:00-23:00 (Figure 6.11A-5). On average he spends 2 days in the camping before he leaves again. Maybe the tourist has a stationary camper located there.

The third longest sequence also corresponds to a 2Axle car (Figure 6.12A-1). This vehicle travels on Fridays on Sundays as depicted in Figure 6.12A-2. He travels from Entrance3 to Camping0 and vice versa between 09-03-2016 until 22-04-2016 (Figure 6.12A-3).

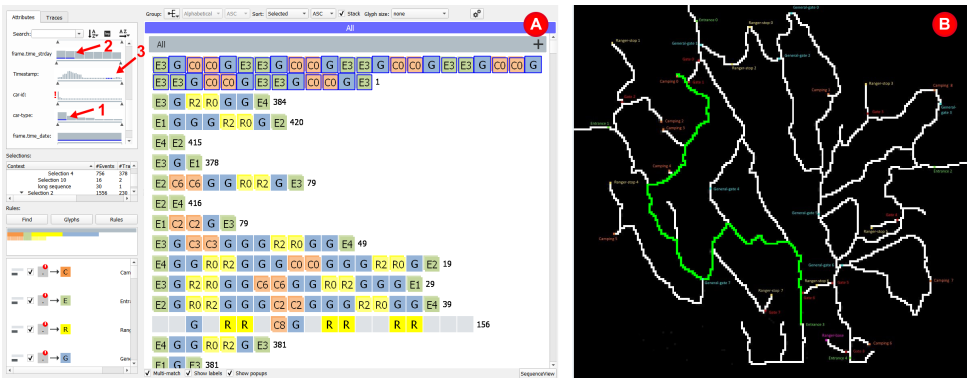


Figure 6.12: A) Long stay of visitor at camping 0. B) Graphical representation of the driven pattern.

Night patterns

In the previous section we studied periodical patterns by looking for repetition in the travel pattern of a single vehicle. To study patterns between vehicles overnight we only consider the traffic between 23:00 and 04:00. This is achieved using the query illustrated in Figure 6.13A. Next, we group the sequences by date (instead of car-id) and color the events by car_type. Now a sequence shows the number of times vehicles have passed through gates per day.

6

Sorting the sequences alphabetically shows that on certain days car type events occur more than others (Figure 6.13B). Selecting the sequences with 3Axle buses for instance shows in the Attribute view that night activity of these vehicles mostly happens during low season (i.e., the period October-April, Figure 6.13B-1). Furthermore, we can use a mini-map to inspect patterns in larger collections of sequences.

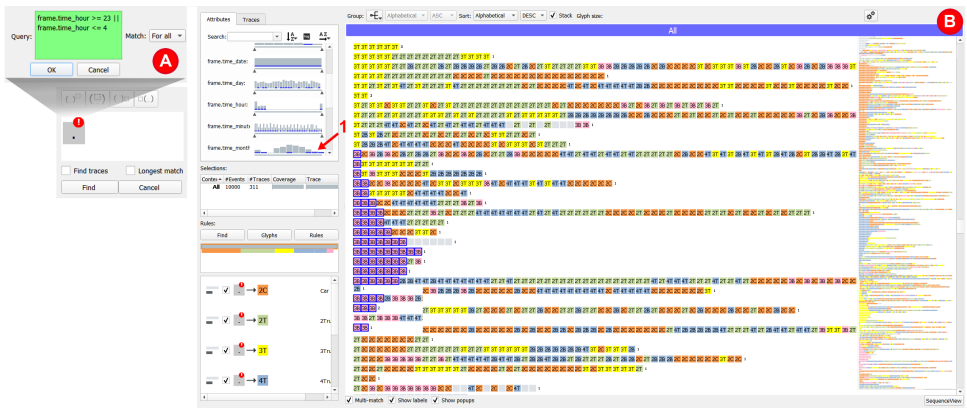


Figure 6.13: A) Query for all traffic overnight. B) Large trucks and buses drive in the night. Few campings are visited. C) Overview of night patterns per day.

6.7. Q3: Unusual Patterns

The third subquestion refers to the detection of unusual patterns. In the previous section we focused on the detection of periodic patterns by exploring the number of times vehicles visited the preserve. Although this gave us new insights, we did not check if there are strange routes inside a single visit. To investigate this closer we first group the events by `car-id` and color the events by `car_type` (illustrated in Figure 6.14).

After clustering the sequences and sorting the sequences alphabetically we see that every vehicle has a few long sequences. The longest sequence of the 2Axle trucks in Figure 6.14B corresponds to the tourist bus in Section 6.6. The longest sequence of 3Axle trucks was unexpected. The Attribute view in Figure 6.14A-1 shows that the truck only visited `Camping1` and `Camping5` on September 8 and 9 (Figure 6.14A-2). Figure 6.14A-3 shows that he only visited the places between 08:00-09:00 and 16:00-18:00.

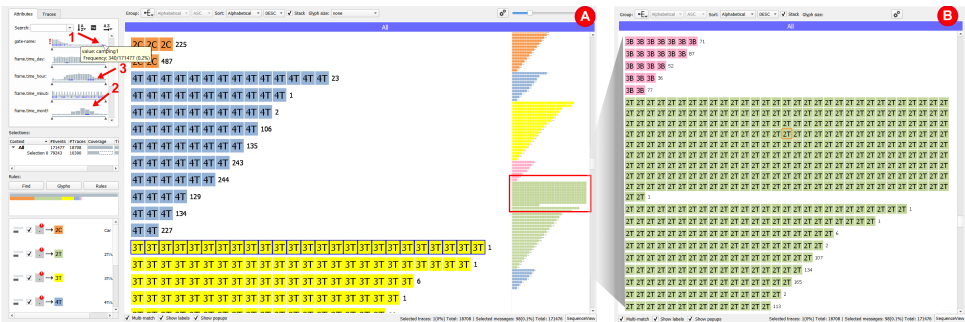


Figure 6.14: A) Frequent and rare car patterns can be discovered by counting how many events are generated per car. B) The mini-map widget shows a long pattern in 2Axle truck sequences.

Unauthorized passage

According to the access matrix in Figure 6.2, only ranger vehicles are allowed to travel through rangerstops. In Eventpad we can easily verify this statement by searching for sequences with rangerstops whose car type differs from ranger vehicles (Figure 6.15A). This reveals 23 cases where 4Axle trucks (Figure 6.15B-2) are going from `Rangerstop6` to `Rangerstop3` to `Rangerstop6` via `Gate6` (Figure 6.15A-1). Inspection of the multivariate data in these sequences in a tabular view shows that the sequences occur between May 2015 and May 2016 (except April, Figure 6.15B-3) between 02:00-05:00 (Figure 6.15B-4) on Tuesdays and Thursdays (Figure 6.15C-5).

Long term visit

The third longest sequence in the data set (Figure 6.16-1) corresponds to a 2Axle car traveling from 06-06-2015 to 20-05-2016 on days other than Wednesday and Saturday (Figure 6.16-2). He travels from `Entrance0` to all the campings (except for `Camping7` and `Camping8`) between 08:00-19:00 (Figure 6.16-6). After approximately one month (Figure 6.16-3) he travels to the next camping (except in the month of April, Figure 6.16-4). According to the last event, he never left the preserve (Figure 6.16-5). The ordering of camping visits is strange and seems random. Maybe he is an ornithologist seeking for a particular bird species.

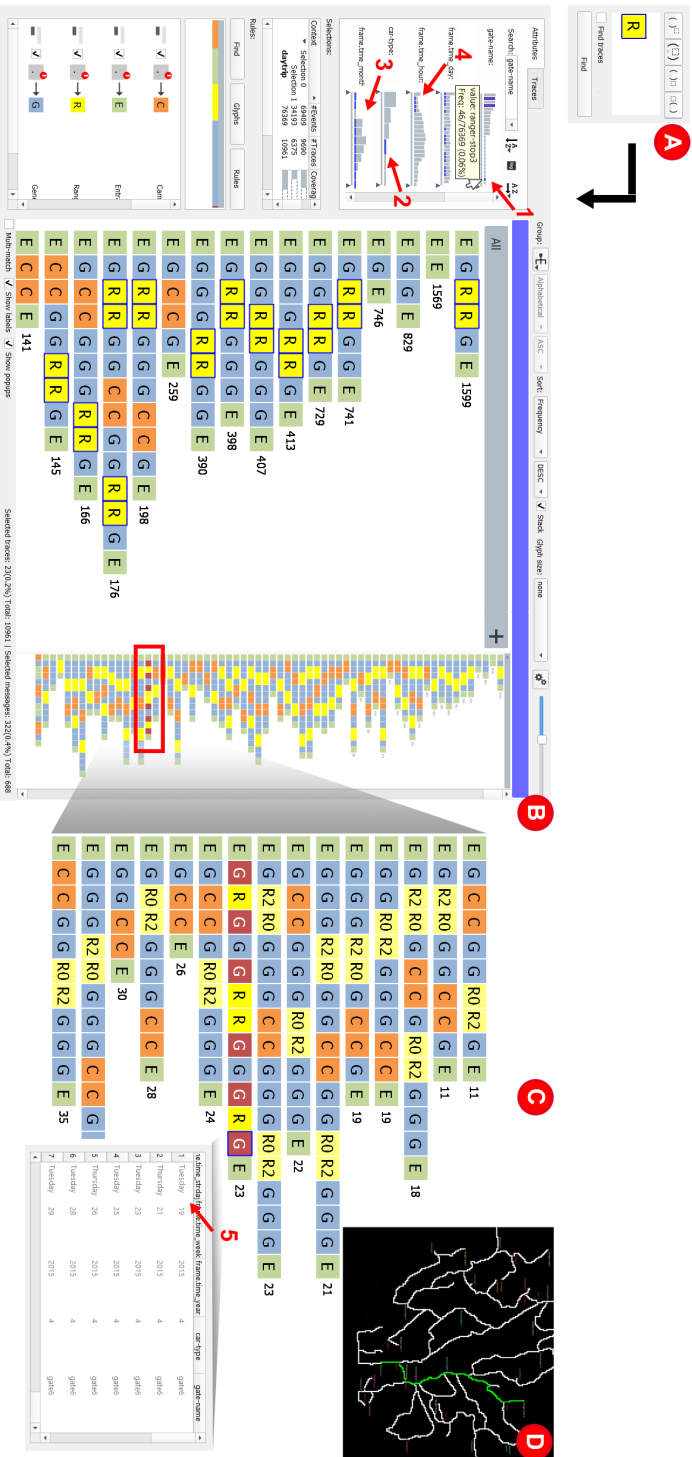


Figure 6.15: A) Search for rangerstops B) Deviating pattern in overview C) Inspection shows the presence of a 4Axle trucks in range routes. D) Graphical representation of taken route.

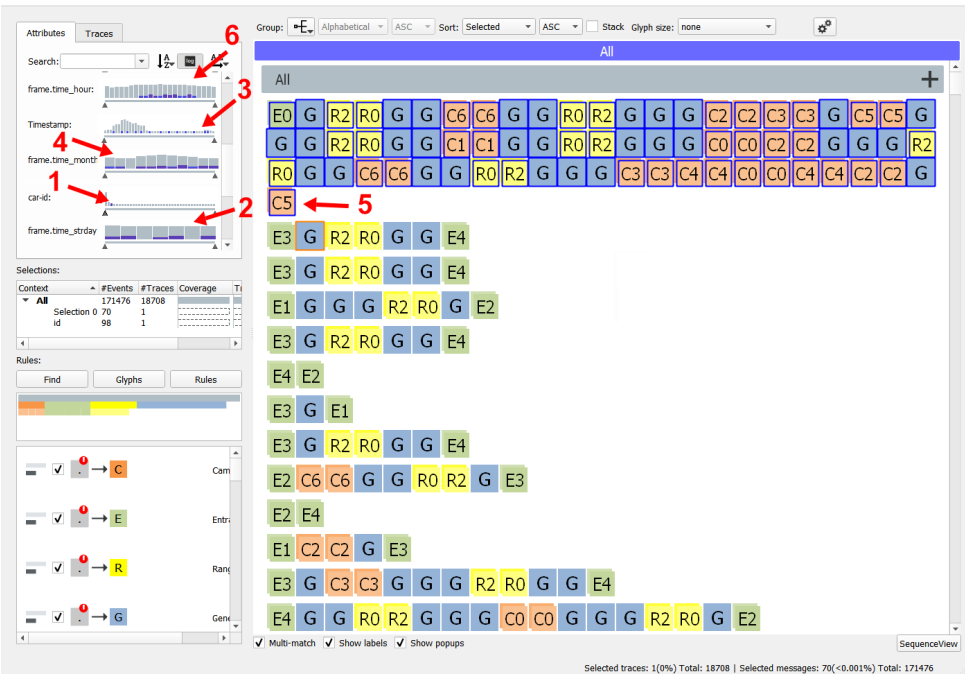


Figure 6.16: Strange pattern of a vehicle traveling through the entire preserve in 350 days. It is unclear why he chose to visit the campings in this particular order.

Traveling truck

Between 12-07-2015 and 04-08-2015 there is a 3Axle truck (Figure 6.17A-1) traveling between Camping6 and Entrance4 (Figure 6.17A). He initially entered the preserve via Entrance2 (Figure 6.17A) and drives only on Tuesdays and Sundays (Figure 6.17A-2). It is unclear why he entered the preserve from Entrance2.

Day trip loops

When extracting day trip patterns in Section 6.5, we noticed that there are sequences with the same entrance and exit. Table 6.3 shows a summary of these “loops”. Inspecting these sequences in greater detail shows that they are caused by 4Axle trucks (Figure 6.18D-1), traveling on all days except Mondays and Fridays (Figure 6.18D-2) between 18:00-22:00 (Figure 6.18D-3).

The sequences can be delivery trucks dropping supplies at entrances for special occasions (e.g., fireworks on the 4th of July). However, Figure 6.18D-4 shows that the time between entering and leaving is at most 5 seconds.

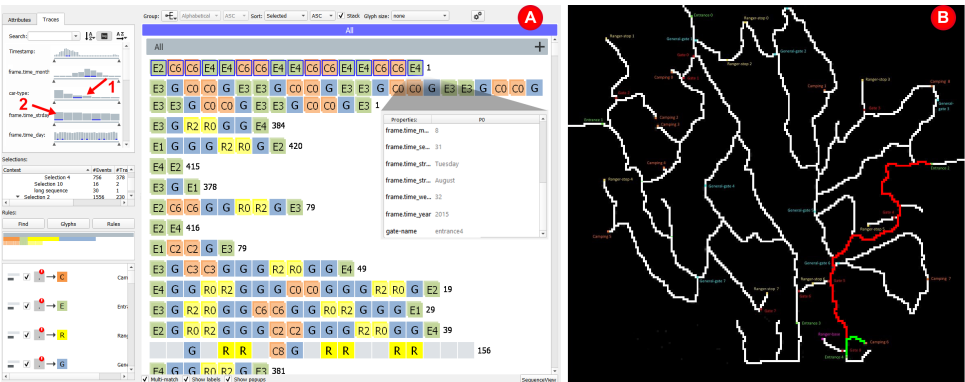


Figure 6.17: A) Long and repeated stays of a visitor in Camping 6. He uses a 3axle truck to drive from and to the camping. B) The route in red is only traveled once by the visitor.

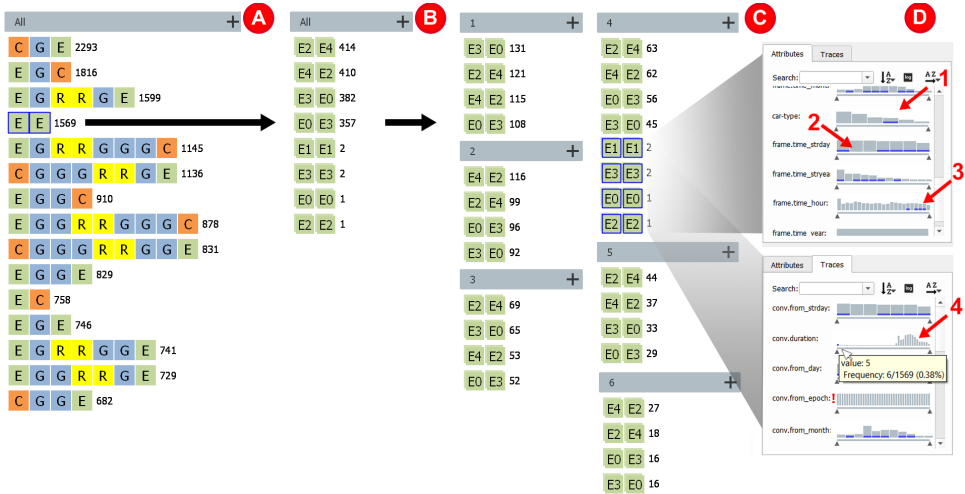


Figure 6.18: A) Extracting direct roads B) Inspecting routes with a more detailed ruleset C) Grouping traffic by car type. D) Histograms showing the location of these sequences in time.

From	To	Duration (seconds)	Arrival Date
Entrance1	Entrance1	5	04-07-2015 22:02
Entrance1	Entrance1	5	23-03-2016 21:06
Entrance2	Entrance2	5	26-06-2015 22:34
Entrance3	Entrance3	6	01-09-2015 20:45
Entrance3	Entrance3	5	18-05-2016 18:10
Entrance0	Entrance0	5	22-10-2015 20:03

Table 6.3: Routes with the same entrance and exit are taken throughout the year. Strangely, the time between entering and exiting is less than 5 seconds.

6.8. Conclusions

In the previous sections we have discovered different daily, periodic, and unusual patterns using Eventpad to gain better insights in desired and undesired sequences in the Lekagul traffic data. Based on these observations we believe that there are three main reasons that are most impactful to bird life in the nature preserve, namely:

- Vehicles on the road between `Entrance0` and `Entrance3` drive too fast. The noise can disturb the wildlife.
- The repeated access of vehicles to unauthorized locations (in the middle of the night) and the presence of systematic travel activity across the entire preserve (e.g., tourist buses) during high-season can prevent wildlife from establishing a proper breeding place.
- The continuous nightly activity of vehicles such as buses and trucks over the entire year can disturb the wildlife.

We have shown the effectiveness of Eventpad to quickly gain insight in the VAST 2017 Mini Challenge 1 data set. The ability to visually encode event properties in sequences using rules enables users to quickly discover patterns inside sequences. Pattern aggregations and selections enable users to study commonalities and differences between sequences while staying aware of high-level phenomena in the data set.



Rapid Reverse engineering of Malware Behavior

This Chapter is based on [44]:

B.C.M. Cappers, P.N. Meessen, S. Etalle, and J.J. van Wijk.

Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics.

In Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec) 2018, pp. 1-8

7.1. Rapid Malware Analysis and Reverse Engineering

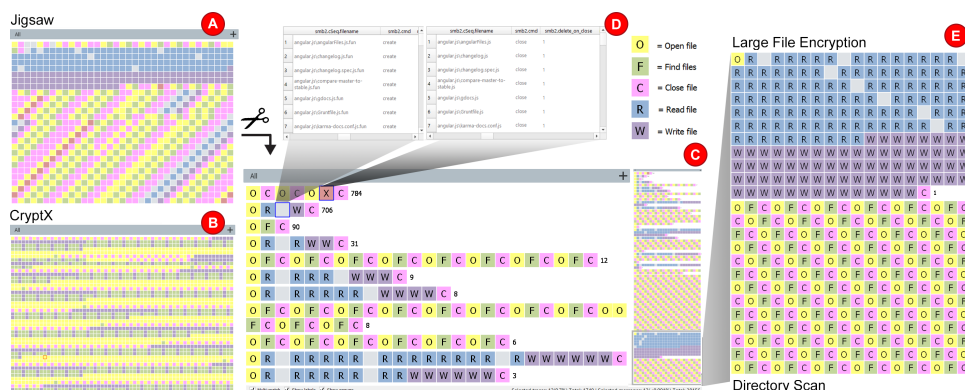


Figure 7.1: Visualizing A) Jigsaw and B) CryptX ransomware activity. C) Partitioning the data by file nesting shows high repetition of file creation and deletion patterns in the mini-map. D) Inspection of protocol data in a tabular view shows the creation of duplicate files with a .fun extension. E) Repetitive “Open-Find-Close” patterns as a result of recursive directory scanning.

Forensic analysis of malware activity in network environments is a necessary yet very costly and time-consuming part of incident response. Vast amounts of data need to be screened, in a very labor-intensive process, looking for signs indicating how the malware at hand behaves inside e.g., a corporate network. We believe that data reduction and visualization techniques can assist security analysts in studying behavioral patterns in network traffic samples (e.g., PCAP). We argue that the discovery of patterns in this traffic can help us to quickly understand how intrusive behavior such as malware activity unfolds and distinguishes itself from the “rest” of the traffic.

In this chapter we present a use case of the visual analytics tool EventPad and illustrate how it is used to gain quick insights in the analysis of PCAP traffic using rules, aggregations, and selections. We show the effectiveness of the tool on real-world data sets involving office traffic and ransomware activity.

7.2. Introduction

The analysis and identification of malware in computer environments is a complex and time-consuming task due to the size and variety of generated network traffic. Even if security analysts are aware of the presence of undesired activity (e.g., ransomware encryption patterns) it is still difficult to efficiently locate this behavior in large amounts of traffic using tools such as Wireshark [63] without a starting point [92, 203]. Automated analysis techniques already greatly assist analysts in finding points of interest. In practice, however, they are often time-consuming to setup or very difficult to tune properly (e.g., managing false positives rates). We need alternative techniques to quickly explore patterns in malicious network traffic.

In this chapter we present the application of a novel visualization technique Eventpad [47]

to apply dynamic behavioral analysis to ransomware execution sequences. Instead of automatically trying to discover patterns of interest, we show how we can quickly gain insights in (un)desired traffic patterns by visually encoding traffic based on environmental knowledge. In this study we show how the analysis technique works, the kind of information it can reveal, and how it enables analysts to quickly study file access behavior using rules, aggregations, and selections. Specifically, our main contributions are:

- a visual analytics approach to forensic analysis of malware traffic enabling users to visually inspect and detect (un)desired patterns of interest using multivariate regular expressions.
- a use case of the EventPad system to malware analysis demonstrating how
 - visualization of execution sequences can be used to gain insight in underlying ransomware mechanics;
 - rules, automated techniques, and user interaction enables users to quickly test, discover, and compare traffic execution sequences.
- the introduction of mini-map functionality and a temporal view to study larger event collections and the frequency of sequential patterns over time.

The chapter is structured as follows. First, Section 7.3 presents related work. Next, we discuss how our visualization techniques are beneficial for the analysis or identification of malware traffic. In Sections 7.5 and 7.6 we provide example explorations on real-world ransomware and office traffic and discuss the limitations of the approach. Finally, conclusions and future work are presented in Section 7.7.

7.3. Related Work

Malware visualization is an extensively studied topic covering a wide variety of techniques in different domains. Eventpad focuses on dynamic malware analysis using Deep Packet Inspection [253]. Analysis techniques that take the source code of the malware into account (i.e., static and hybrid malware analysis) are considered out of scope. For more information about these techniques, we refer to the handbook of Sikorski et al. [279].

The visualization literature with respect to *dynamic* malware detection can be grouped in two main categories, namely malware *discovery* versus *identification*. We start with an overview of common malware visualization techniques, discuss current limitations, and how we address these in Eventpad. Finally, we describe different visualization techniques that have been applied in the areas of Deep Packet Inspection and ransomware visualization.

7.3.1. Malware discovery

Malware discovery is the task of extracting samples from (network) environments that show signs of undesired/intrusive behavior. Shiravi et al. [274] made an extensive overview of different security visualization techniques that have been used for:

- *detection*: visually spot anomalies using node-link diagrams, parallel coordinate plots and pixel visualizations [80];
- *correlation*: study patterns in IDS alerts to enhance decision making [203]; and
- *assessment*: apply *root-cause* analysis through semantic zooming [66] and user-interaction [347].

Zhang et al. extended this survey [353] by studying how current visual encodings are effective for detection, correlation, and assessment tasks [229]. We believe however that systems for network monitoring and digital forensics should not be limited to a single task or visual encoding, as the set of user tasks can change during analysis [134]. In Eventpad we aim for an alternative approach where analysts can interactively define themselves which data attributes should be represented in what way to serve their task at best.

With ILAB, Beaugnon et al. [22] already illustrated the value of human interaction to incrementally label data instances for supervised intrusion detection models. Systems such as KAMAS [270, 327] also use rules to search for patterns in call sequences, but limit their sequential analysis to only this attribute. In Eventpad we show how incremental labeling combined with unsupervised clustering and alignment techniques are effective for the discovery of patterns in multivariate network traffic.

7.3.2. Malware Identification

7

Gaining insight in the working of malware is crucial for understanding:

- what (kind of) systems are affected by the malware;
- the type of services or data the malware is interested in; and
- what countermeasures can be used to prevent this type of software from being executed in the future.

Eventpad assists analysts in the identification of (malicious) execution sequences by enabling users to define and search for patterns of interest using rules. Visual comparison of the packet details between the found sequences enables analysts to see whether the sequences show overlap and can be related to existing malware families or applications.

Wagner et al. [326] provided an extensive analysis of visualization techniques to visually compare malware samples, study samples individually, and summarize collections of malware together. Popular systems in these categories are CantorDust [79], Nataraj et al. [231], and Anders et al. [9] respectively. In their taxonomy Wagner et al. identified three main limitations of current systems and challenges for the future, namely

- Incorporating expert knowledge through interaction;
- Intertwining analytical methods with visualization; and
- Bridging gaps between forensics and classification through *rule generation*.

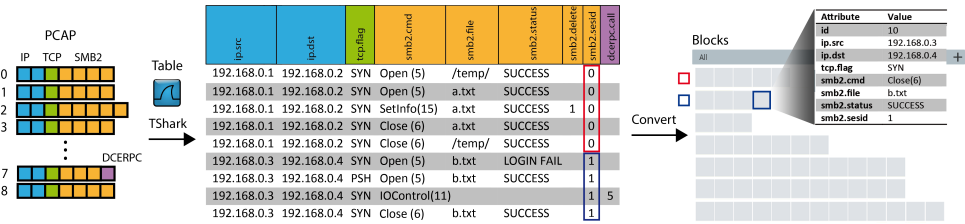


Figure 7.2: Deep Packet Inspection of network packets obtained using `tshark`. Parsed traffic is grouped by an attribute of choice and represented as series of blocks.

In Eventpad we tackle these limitations by enabling analysts to define new patterns of interest using conditional formatting and rewrite rules. Tight coupling between automated methods and user input is achieved by interactively visual encoding packet sequences as *blocks*. The third limitation refers to the inability of malware analysis systems to test the effectiveness of newly discovered rules and signatures for the analysis and classification of other malware samples. In Eventpad rules can be added and removed dynamically throughout data exploration and exported for future analysis tasks.

7.3.3. Deep Packet Inspection

Eventpad uses Deep Packet Inspection (DPI) to study user behavior in network traffic. Various systems have been proposed to visualize DPI data for the detection of Advanced Persistent Threats [280]. Systems like Wireshark [63], SNAPS [45], and CoNTA [46] already support the analysis of traffic at application level, but the provided search mechanisms and visualizations do not support the comparison and analysis of sequential patterns in network traffic. However, Camiña et al. [42] showed that for the detection of for instance masquerade attacks, the analysis of sequential patterns is crucial. We show how we can extend Wireshark to enable sequential analysis using Eventpad. With respect to ransomware analysis, Reuille et al. [262] analyzed the spreading of the cryptolocker virus by visualizing OpenDNS traffic. Krzysztof et al. [183] also illustrated the value of visualization to gain faster insights in larger collections of traffic.

In summary, current malware visualization techniques use static visual encodings to either focus on the detection of specific malware behavior or the identification of it. Systems that provide interaction and analytics methods provide little support for incorporating expert knowledge in the system or feeding new discovered patterns in these analytical methods. Eventpad enables interactive visual encoding through rule generation to dynamically detect signs of new malware and to verify the presence of existing ones.

7.4. Eventpad

Eventpad is a dynamic behavioral analysis tool designed to study sequential patterns in network traffic. Without loss of generality, we assume that a network packet (or *event*) has a timestamp and belongs to a particular *sequence* (also known as a conversation, session, case or trace).

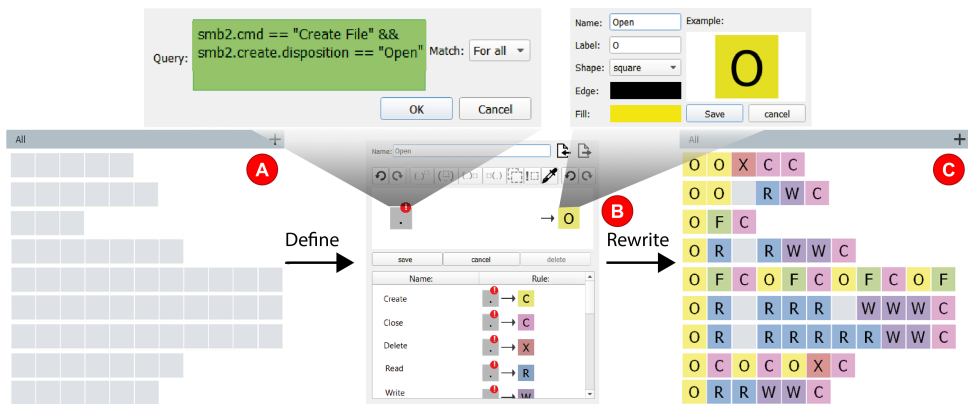


Figure 7.3: A) Users can visually encode sequences according to attributes of interest by constructing one or more rules. B) Rules are constructed using multivariate regular expressions. Users can design their own blocks to highlight points of interest. C) Event collection after rule rewriting.

In Eventpad we visualize network traffic as a list of block sequences where sequences correspond to *network conversations* and packets correspond to *blocks* (Figure 7.22). Initially every packet is represented by a gray block. Users can create *rules* to replace block sequences by a new (custom styled) one. Rules can be used for three purposes:

- *discover* patterns by visually encoding packets according to attributes of interest;
- *test* the presence of (un)desired behavior through pattern matching; and
- *guide* automated techniques in discovering patterns by defining new (higher level) concepts.

Similar to regular expressions, operators such as sequential composition, iteration (0 or more times), and choice are provided to construct more complex replacements (Figure 7.4A). Figure 7.3 shows an example of the rule construction interface.

Double-clicking a block in the interface results in a popup where users can add Boolean constraints to the chosen block (Figure 7.3B, similar to Wireshark). Only blocks whose attribute and values match the Boolean constraint are replaced by the rules right-hand side. For a more formal model of the query mechanism, we refer to the work of Cappers et al. [47]. Multiple rules can be applied after one another to enable incremental rewriting of the traffic. This is also illustrated in Figure 7.4B.

The rules enable users to highlight and visual encode traffic properties that are of interest. Automated techniques such as clustering and alignment in turn can use this labeling to discover patterns between packet sequences. Clustering enables users to study pattern frequencies, whereas alignment can detect overlap between similar sequences using Multiple Sequence Alignment [31]. Figure 7.5 shows a schematic overview of all operations that can be applied to the Eventpad system.

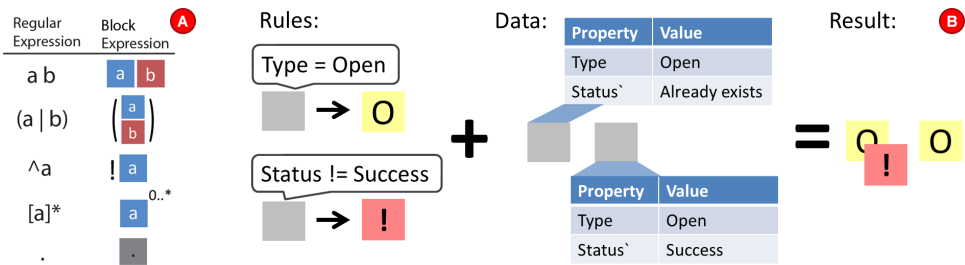


Figure 7.4: A) Block expression use operators that are similar to regular expressions. B) Example where multiple rules can apply to the same block.

Eventpad consists of four main components.

- The Sequence view (Figure 7.6A) visualizes conversations as sequences of blocks. The minimap on the right (Figure 7.6B) enables the discovery of patterns over larger data collections. Selections can be stored using a Context view (Figure 7.6D);
- The Rule view (Figure 7.6F) shows the ordering in which rules are applied to the data. The icicle plot above (Figure 7.6E) shows how much of the data is affected by every rule;
- The Attribute view (Figure 7.6C) shows an overview of all event properties as histogram widgets. The histograms are highlighted whenever one or more blocks are selected. The histogram widgets enable users to inspect overlap in properties between two or more blocks;
- The Line chart (Figure 7.6G) plots event frequencies over time to detect temporal patterns such as bursts, drops, and periodicity. The colors in the chart show when and how often rules have been fired over time.

7.5. Use cases

To illustrate the effectiveness of the analysis tool for digital forensics, we applied Eventpad to the analysis of file access behavior in a real office network. For a better overview of how the system is used in practice, we refer to the supplementary material¹. The system² and recorded ransomware samples³ are also available for download.

¹<https://youtu.be/g4brXOtPELI>

²<http://www.event-pad.com>

³https://security1.win.tue.nl/doku.php?id=artefacts#data_sharing

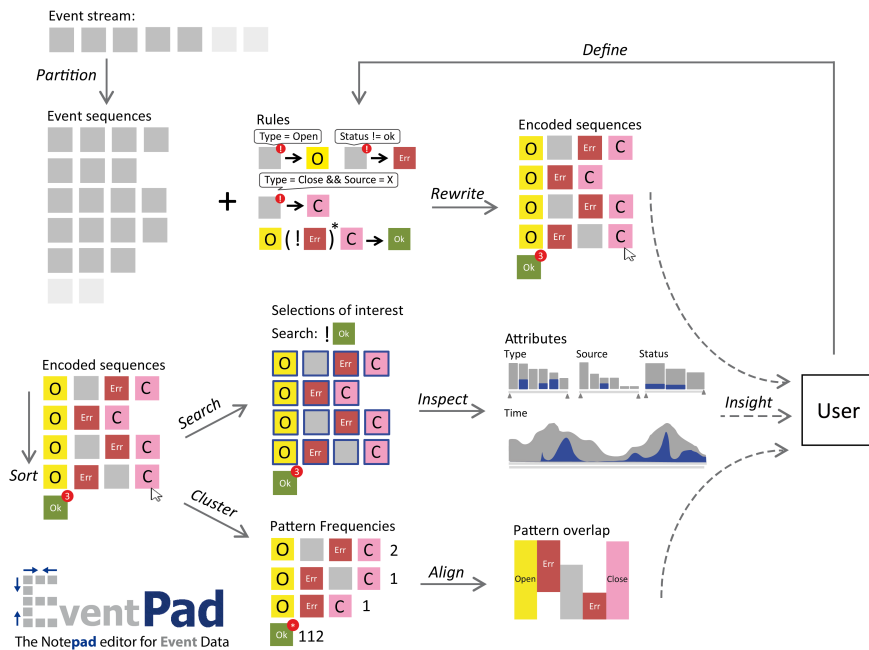


Figure 7.5: Workflow scheme of Eventpad. Simplify event sequences by incrementally replacing block sequences using rules. Compare execution sequences by applying clustering and alignment techniques to the rewritten data. Inspection of attributes in overlapping sequences enables analysts to identify (un)desired behavior in the traffic. New insights can directly be incorporated in the analysis by defining new rules throughout exploration.

7.5.1. Problem statement

Over the last few years ransomware attacks have become an increasing threat to society, with the primary aim to earn money by captivating (critical) resources. The largest class of these viruses use file encryption to achieve their goal [252]. Recent occurrences of WannaCry and Petya [219] affected over 300.000 computers worldwide causing millions of financial damage by encrypting network shares. The faster we can locate and identify their underlying mechanics, the better.

Traditional behavioral analysis tools discover ransomware by analyzing entropy changes in network traffic [288]. These techniques however suffer from high false positive rates, since entropy alone cannot distinguish between legitimate versus ransomware file encryption. Together with Mulder et al. [228] we have extended entropy-based detection in Samba shares by also incorporating file mutation patterns (e.g., read, write, open, close, delete) in the analysis. A significant challenge however was to discover these patterns in network samples using tools such as Wireshark. This involved manually tracking file identifiers and stepping through captured network traffic for every connection. In this use case we demonstrate how we can use Eventpad to reduce days of manually reverse engineering file access activity to just a few hours using visual analytics.

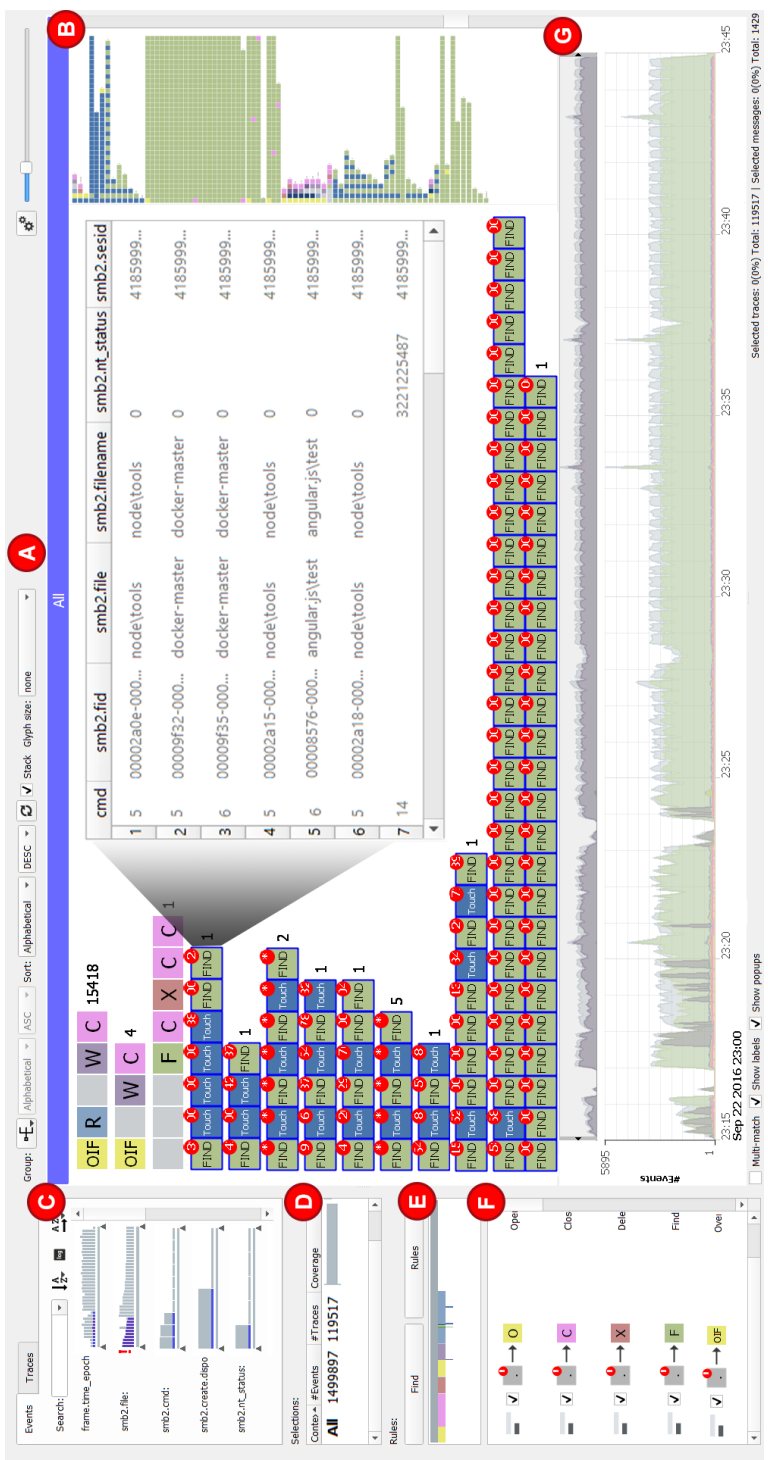


Figure 7.6: Graphical user interface of the implemented prototype and components: A) Sequence view represents sequences as a series of blocks. B) The minimap shows patterns over larger amounts of data. C) The attribute view shows trends and patterns in protocol fields. D) Selections are stored in filters throughout exploration. E) The icicle plot shows the coverage of the applied rules. F) The Rule view shows a list of applied rules during exploration. The ordering for rewriting is controlled via drag and drop operations. G) A line chart can be enabled to study the activation of rules over time.

7.5.2. Experimental setup

To study ransomware activity, we setup a Honeypot and built a detector that can protect Samba shares by passively observing the network traffic [228]. This lab consisted of four victim virtual machines, each with a different version of the Microsoft Windows operation system, which were connected to a Samba share on a virtual machine running Windows Server 2012. The Samba share contained a large collection of files, loosely based on the office type files found on the network shares at the university and the source code repositories of a number of software projects. All these machines were in a virtual network connected to the Internet, since some malware will not run if it cannot connect to specific command and control servers [279].

All network traffic in this virtual network was captured using `tshark`. Before and after the experiment all virtual machines were reset to a verified snapshot. The malware in the experiment was captured by the university, found on TheZoo [232], or found in spam e-mail. Every sample is executed on the victim machines to ensure encryption of the share. Services were stopped once the encryption phase of the samples was over.

Using our setup we captured activity traces from two common ransomware families: `CryptX` and `JigSaw`. We also captured some traffic from the university network shares to test if these ransomware samples were present there.

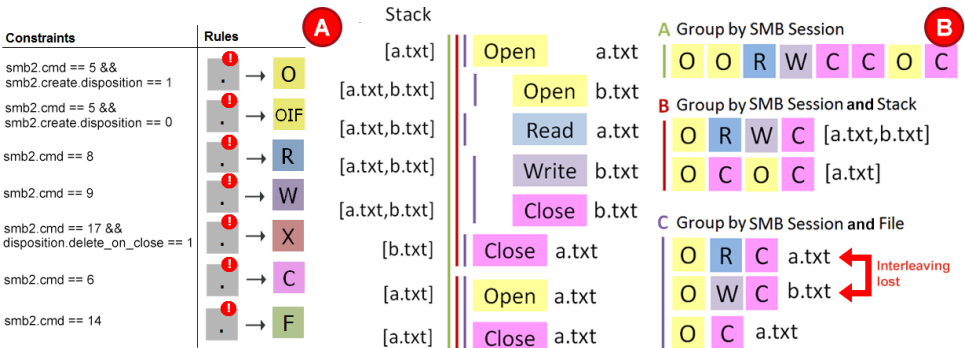


Figure 7.7: A) Highlight rules that are initially applied for the analysis of office traffic. B) Grouping the traffic by a) `smb2.sesid`, b) file nesting, and c) `smb2.fid` shows how different patterns can be discovered.

7.5.3. Partition Strategies

Although the sequence of opening, reading, and closing a file in general may not be suspicious, they can be considered malicious when executed by a particular user and/or moment in time etc. The way we group packets into sequences therefore determines the type of patterns that we can discover. This is also referred to as *context* [46].

Whenever a file or directory is opened, a Samba request is sent. Directories are closed when all files in that directory are closed. This results in a nesting pattern of Open and Close

To ensure that this ordering is preserved, we implemented a stack-based session based on file nesting. Each time a file is opened, the corresponding filename is pushed to a stack. The name is popped from the stack upon encountering its corresponding close request. Figure 7.8 shows the effect of traffic clustering when ransomware traffic is grouped by session, file nesting, or file id. The frequency of every pattern is shown at the end of every pattern.



7.5.4. Ransomware

Jigsaw

- `FILE_OPEN` If the file already exists, return success; otherwise, generate an error.
- `FILE_OVERWRITE_IF`: Overwrite the file if it already exists; otherwise, create one.

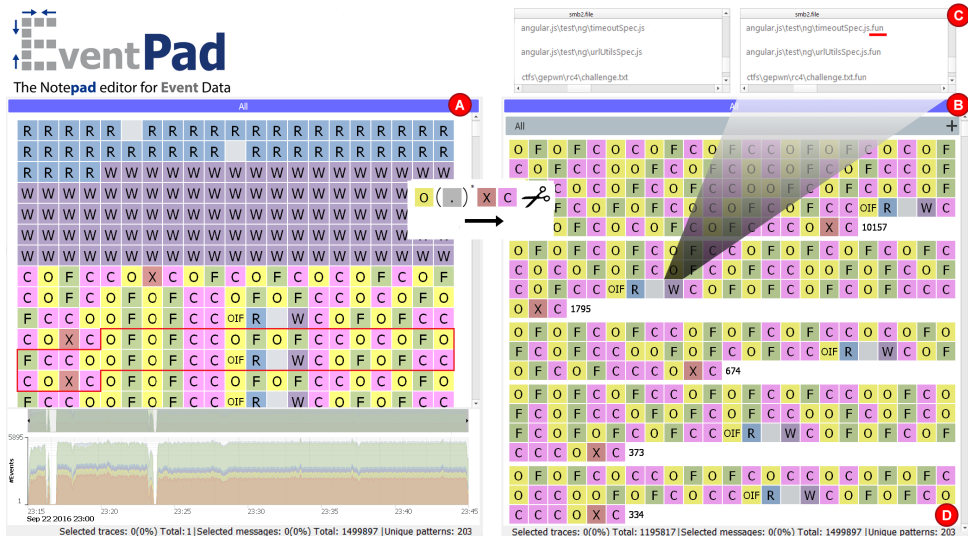


Figure 7.9: A) Repetitive patterns in the Jigsaw traffic. B) Clustered open close patterns show high similarity. C) Inspection of read and writes shows that data is copied to files with `.fun` extension.

These are marked by a yellow “Overwrite IF” and “O” blocks respectively. The attribute widget `smb2.nt_status` in Figure 7.6C shows that the Jigsaw virus did not produce any errors when generating the requests.

7

After applying the highlight rules in Figure 7.7A, the minimap shows repetitive patterns in the traffic (Figure 7.1A). Closer inspection of these patterns shows that at some point files are repeatedly deleted (Figure 7.9A, marked by the “X” block). We can also see that the “OIF” requests only occur before files are read and written. Inspection on the line diagram shows that the pattern repetition started around 23:15 (Figure 7.9A line chart).

To study the frequency of the discovered pattern we constructed a rule that “cuts” the data upon encountering file deletion blocks (scissors, Figure 7.9). This resulted into 1,195,817 sequences as illustrated in Figure 7.9D. Clustering the sequences however shows a lot of similarity between the different sequences (Figure 7.9B). Hovering the mouse over the “read” block of the clustered sequence shows a list of all files that have been accessed in the exact same way (Figure 7.9C). Grouping the data with our stack-based session approach and clustering the data reveals three main patterns in the traffic (Figure 7.1C):

- The creation of a new encrypted file `X.fun`,
- The deletion of the original file `X`, and
- Directory traversal using repeated Open Find Close patterns

This enables us to see that the Jigsaw virus first creates a copy of the target file before it starts encryption. After the encryption, the original file is deleted instead of overwritten. The attack is flawed in the sense that the original file could still be recovered from disk using Disk Recovery tools [115].

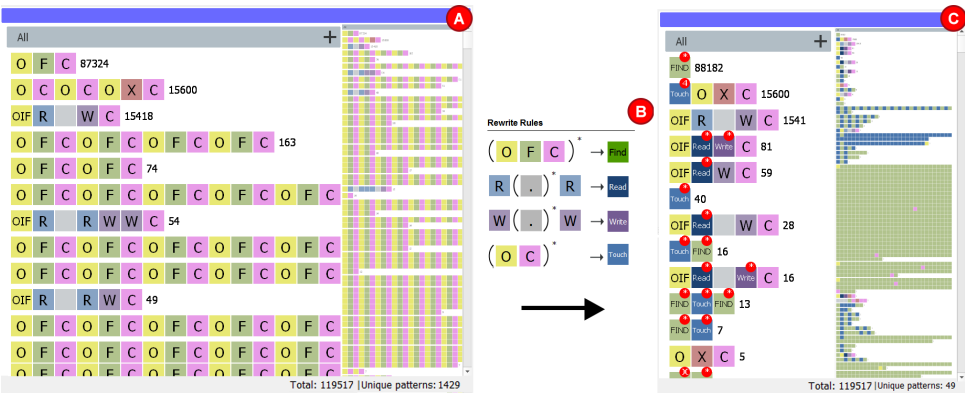


Figure 7.10: Discovering patterns in Jigsaw traffic by capturing repeating access patterns in rules. A) Reduction from 1429 to 49 unique patterns. B) Rule set to compress repeating find, open, and close behavior. C) Resulting traffic after rewriting.

In order to reverse engineer the mechanics of the virus, we construct rules to capture repeated block patterns to higher level concepts. Figure 7.10B shows the rules that are applied to simplify the traffic. The first two rules compress repetitive occurrences of open and close patterns. The frequency of these patterns depends on the file depth in the directory structure. Since the number of sequential read and write patterns depends on the size of the requested file, we also create a rule that compressed these sequences in a single block. After compression, the number in the upper right corner of the blocks shows how many blocks are contained in the new block.

Applying alignment on the remaining patterns revealed that overlap between these file access behavior is large (Figure 7.11A). Figure 7.11B shows the resulting regular expression that captures the Jigsaw traffic.

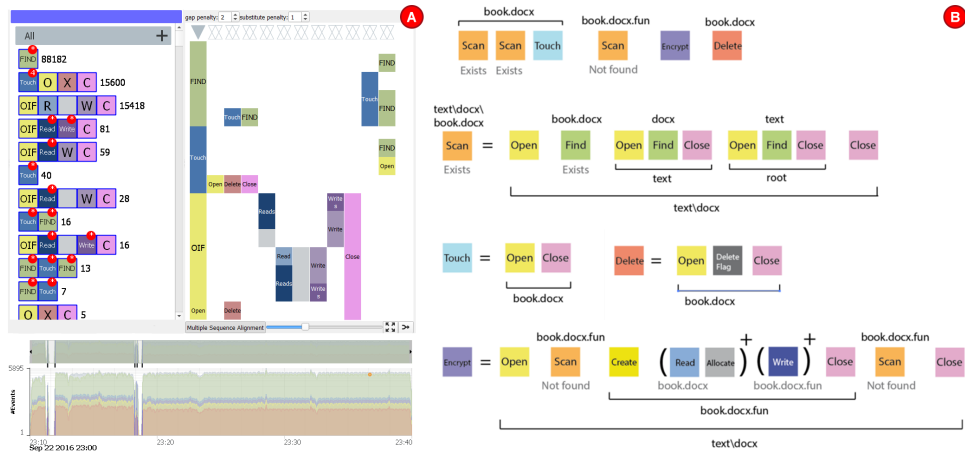


Figure 7.11: Discovering patterns in Jigsaw traffic by capturing repeating access patterns in rules. A) Reduction from 1429 to 49 unique patterns. B) Rule set to compress repeating find, open, and close behavior. C) Resulting traffic after rewriting.

CryptX

For the analysis of Cryptx we start with the rule set as constructed for the Jigsaw virus (Figure 7.7A). Inspection of the line diagram shows that read/write bursts started between 20:19 and 20:35. In contrast to Jigsaw we do not see delete requests.

After applying the rules, the icle plot shows that for some Samba packets multiple rules applied at the same time (Figure 7.12A). Closer inspection in Wireshark showed that these packets consisted of multiple Samba headers (Figure 7.12B). Figure 7.12C shows examples of *compound* requests that represent the opening of a directory and finding files in that directory encoded in one packet.



Figure 7.12: A) The icle plot shows that some CryptX packets are both Open and Find requests. B) Inspection with Wireshark. The minimap (C) and line chart (D) show that compound requests are sent before directory traversal and outside the encryption phase. E) Grouping the traffic by `smb2.filename` shows that files are overwritten. F) During encryption files are opened and closed sequentially.

Compound Samba requests in general are valid with respect to the protocol specification as they have been introduced for efficiency reasons. There are however known bugs in practice with these constructs [36–38]. In addition, Samba intrusion detectors without DPI may actually be unaware of multiple headers in Samba traffic. The evaluation of viruses that use compound requests to masquerade file reads and writes in practice is left for future work. After searching for compound requests in the data the line chart shows that these commands were sent at the start and end of the burst period (Figure 7.12D).

To study the overall duration of sequences, users can enable an arc diagram. This will draw an arc from the start to the end of every sequence (Figure 7.12D). The larger arcs correspond to the opening and closure of directories whereas the small arcs correspond to file access.

Grouping the traffic by `smb2.file_name` shows that repeated read and write activity only

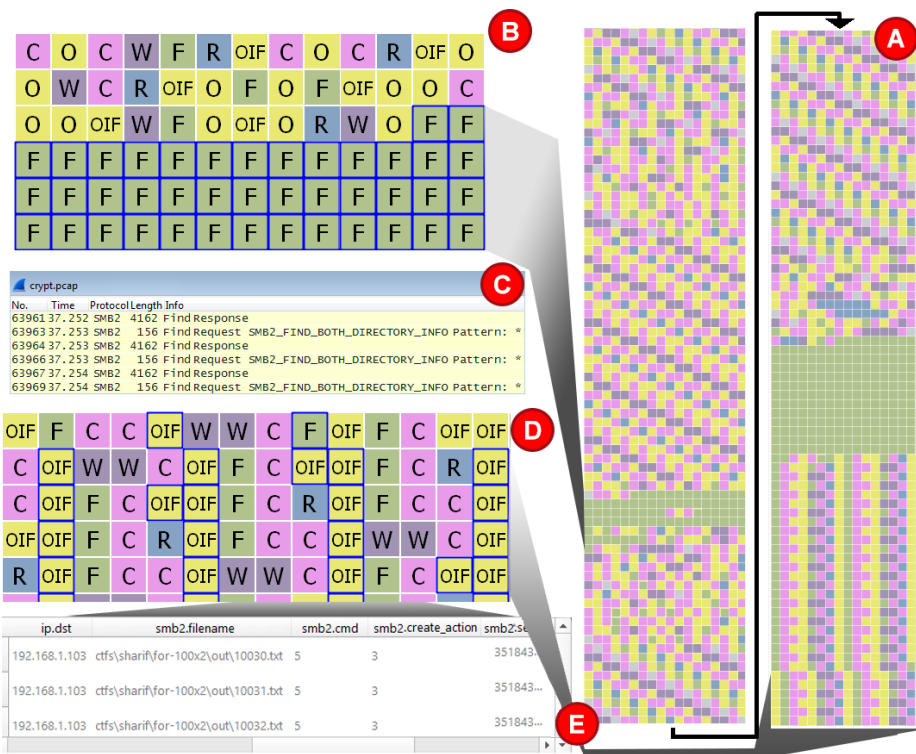


Figure 7.13: Studying sequential Cryptx file access patterns (A). B) Example of a recursive directory scan before encryption C) Wireshark detailed view. D) Detailed overview patterns E) Tabular view shows files are accessed alphabetically.

happened during the burst period (Figure 7.12E). Zooming in on the line chart shows smaller arcs indicating that files were opened and closed in short periods in time (Figure 7.12F). Inspection of the burst period in the minimap shows that the burst consists of a large number of find requests followed by repetitive file read and write patterns (Figure 7.13). The find requests in Wireshark show that the virus, in contrast to Jigsaw, first recursively searches directories for files before the encryption burst starts (Figure 7.13A). Inspection of the burst shows that files are encrypted sequentially (Figure 7.13D) and are traversed alphabetically (Figure 7.13E).

7.5.5. University Traffic

We studied the patterns in the ransomware samples and compared them to recorded traffic from a university that has been struck earlier by ransomware. The main question was whether there was still ransomware activity inside the network.

For the analysis of university traffic we recorded internal network traffic of 20 hosts over a period of a month. Out of this 94GB of traffic we extracted all smb2 meta-data using the `tshark` protocol dissector. This results in the analysis of approximately 14,000,000 packets

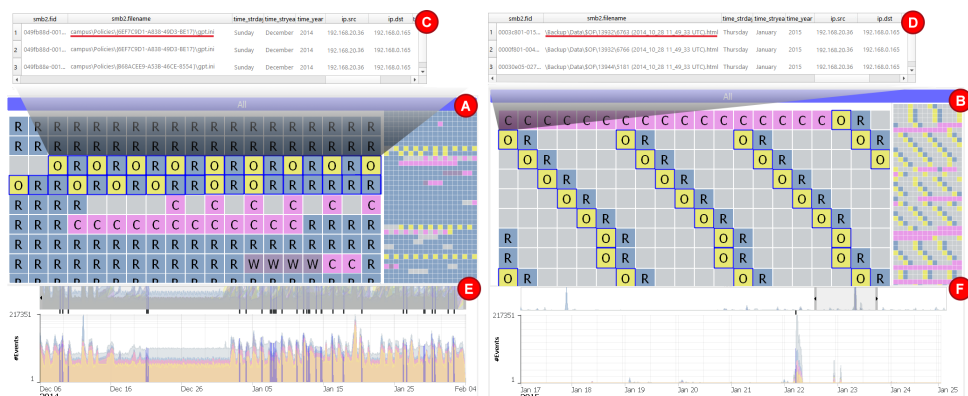


Figure 7.14: A) Access patterns to Microsoft Policy files (C). B) Access patterns of a user backup (D). Note that files are accessed in parallel (not sequentially). E) Policy files are accessed on multiple periods in time. The line chart shows patterns on a logarithmic scale. F) Backup burst happened on January 22nd.

with over 100 protocol fields. In order to reduce the large packet count, response values were merged in the corresponding requests based on their `smb2.msg_id` and session. In addition, file access to print services such as `spoolss` were excluded from the analysis.

We initially study the traffic by loading the rule set that we constructed with the ransomware analysis. The line chart in Figure 7.14 shows the presence of a traffic burst on January 22nd (Figure 7.14F). Grouping the traffic by `smb2.sesid` and inspecting the traffic shows that a user is reading files from his backup directory (Figure 7.14D). In contrast to the ransomware viruses, we can see that Open and Close packets are not alternating, but happen in bursts showing that in the university traffic files are accessed in parallel rather than sequentially. Figure 7.14A also shows this for traffic outside the traffic burst.

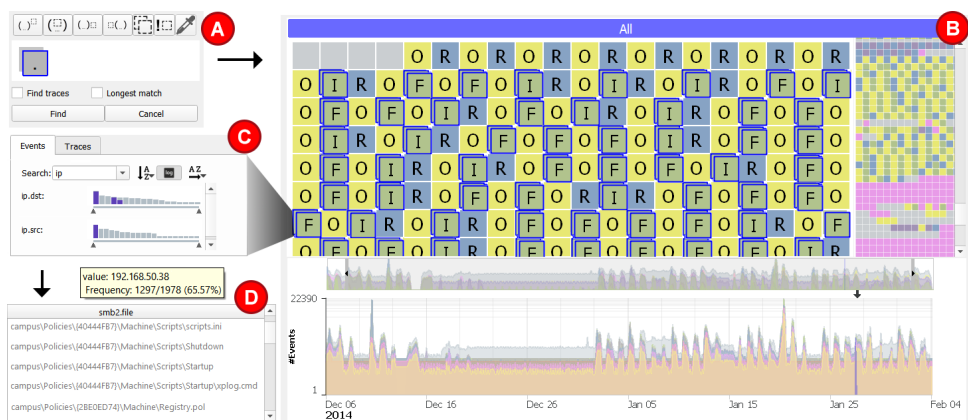


Figure 7.15: Searching for Cryptx virus in University traffic. A) Search query for compound requests. B) Discovery of compound requests. C) Requests are originating from one source. D) All blocks access Microsoft Network Policy files.

To check for signs of the Cryptx virus we start searching for compound requests using Eventpad's find dialog (Figure 7.15A). This revealed a burst of Open-Find compound requests similar to Cryptx (Figure 7.15B). The main difference was that files after the burst were only read. Inspecting the Attribute view shows that all compound requests were sent after the burst by the same IP address (Figure 7.15C). Inspection of the file names revealed that they were all related to the reading of Microsoft Network Policy files (Figure 7.15D).

To study the access patterns to these files in greater detail, we search for all packets that involve policy files. Figure 7.14A highlights the packets as a result of the search. Figure 7.14E shows that the pattern occurred several times in the data set. Filtering out the rest of the traffic, we can see that these policy files have similar access patterns (Figure 7.16A).

Although the reading of Microsoft Network Policy files is necessary to determine authorization, we know that the hosts involved in the recording are not authorized to change these policies. To verify this hypothesis, we search for conversations containing write requests (Figure 7.16A). Grouping the traffic by `ip.src` shows that several IP addresses that were violating this constraint (Figure 7.16B). Figure 7.16D shows that these users were modifying `RemoteInstall`, `Logoff` scripts and `Registry` files to gain access to the share from a virtual machine.

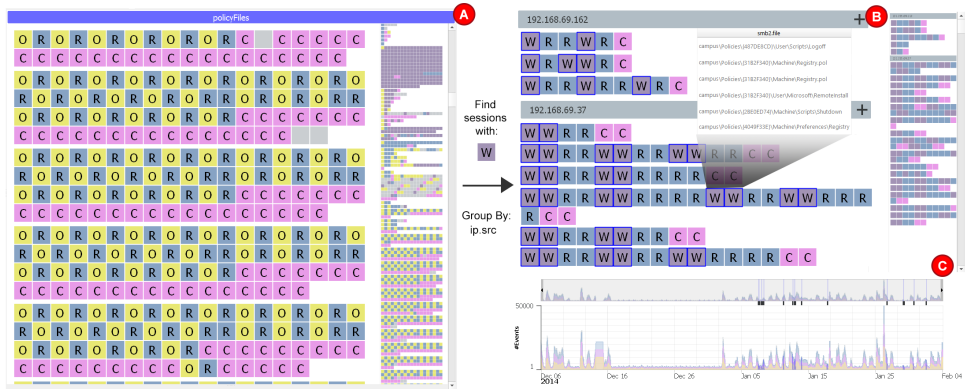


Figure 7.16: Extraction of policy files. A) Overview policy access patterns B) Searching for file modification requests and grouping the traffic by `ip.src`. Discovery of unauthorized IP addresses modifying policy files. C) File modification happened throughout January.

7.6. Discussion

The use cases illustrate how we can use Eventpad to quickly gain insight in user activity by analyzing file access behavior in PCAP traffic. Users are enabled to simplify, focus on, and detect patterns inside network conversations by defining rules. Automated techniques in turn use the labeling to assist users in discovering patterns between different conversations. Inspection of packet attributes enables users to see whether these patterns share overlap in other attributes after which rulesets can be refined. There are several reasons for the rapid discovery of patterns in Eventpad:

- Constructed rules sets are efficient to evaluate as they are based on regular expressions.
- Users do not need to have a full specification of what “good or bad” behavior looks like, but can incrementally obtain results by visually encoding knowledge that they have about their data.
- Unsupervised automated methods in the background use the constructed rules sets to provide nontrivial insights.

Besides the creation of traditional black or whitelisting signatures, analysts create rules to assist them discovering unknown patterns and focusing on parts that are relevant for their investigation. The constructed rule set can be evaluated fully automatically on new (incoming) data. By saving and loading rulesets analysts can quickly verify the presence of malware in other data sets.

Although the tool quickly enables analysts to isolate areas of interest in large network samples, there are threats to validity. First, we analyzed ransomware activity by extracting file access behavior from network traffic. Ransomware viruses that do not focus on repeated (network) file encryption (e.g., UIWIX [307]) cannot be detected with this approach. Although Eventpad in general works on any event data, other data sources are required to make the technique useful for these types of malware.

Second, based on the Samba signaling alone we cannot distinguish between user-initiated file encryption versus ransomware encryption. Although the patterns in the use cases showed clear differences between ransomware and university traffic (sequential versus parallel activity), this does not hold in general. Additional (automated) anomaly techniques are required to verify if the observed patterns indeed correspond to file encryption.

Third, the Eventpad system suffers from a “cold start” problem in the sense that users must already be aware of properties that are of interest. Naively applying clustering or alignment without a rule set results in no insights (underfitting) whereas creating too many rules can result into “noisy” patterns (overfitting). Also to deal with phenomena such as concept drift [309], rulesets need to be maintained by users to ensure the discovery of new viruses in the future.

7.7. Conclusions and Future work

In this chapter we demonstrated a visualization tool to support rapid and cost-effective analysis of network traffic analysis and malware activity. We have shown the effectiveness of the system in real-world ransomware and office traffic. Our tool shows how visualization can be used to quickly gain insight in malware and network activity by combining data reduction and automated techniques in one interface using rules, aggregations, and selections.

Future work will consist of integrating Mülders automated network detection technique with the Eventpad system. In addition, the Eventpad system will be expanded to make real-time monitoring of network traffic and evaluation of temporal constraints feasible.



Conclusions

8

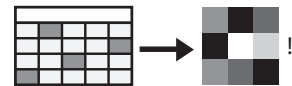
8.1. Overview

During the design of the prototypes and application of the systems in practice, we have gained new insights on how to discover patterns and anomalies in large event collections. In the next paragraphs we briefly summarize the conclusions of the previous chapters and discuss the strengths of each system, all aiming at answering our research question:

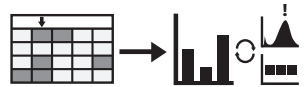
“How can we use interactive visualization techniques and automated methods to discover relevant patterns in large event collections?”

In Section 8.2, we elaborate further on our research question by reflecting on our contributions and providing guidelines for the effective use of automated techniques and visualization of multivariate event logs. Finally, suggestions for future work and final conclusions are provided in Section 8.3.

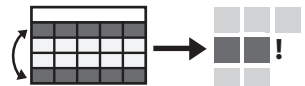
SNAPS: Monitoring Multivariate Event Collections In Chapter 3 we presented a novel approach for domain experts to discover anomalies in network traffic by combining Deep Packet Inspection, machine learning and visualization into one coherent system. The ability to create multiple selections in parallel enable experts to drill down or to focus on specific entities, while still maintaining an overview of the state in the network. The Time view enables experts to detect patterns and trends over time, while the Pixel, Attribute and Lens view enable experts to detect outliers. Furthermore, the ability to train and refine classifiers on multiple selections of interest makes the approach flexible enough to be optimized for very specific environments. We have shown the effectiveness of SNAPS on two real-world data sets. Since the approach only relies on the structure of parse data in general, the proposed method is suitable for application in other domains.



CoNTA: Contextual analysis of Anomalous Events Chapter 4 presented a novel approach for domain experts to explore large message collections using interaction and automatically generated alerts. The ability to interactively switch from traffic-level overviews to message-level details enables experts to investigate the relationships between high-level traffic phenomena and low-level message fields, while staying aware of other concepts, such as conversations and sequential patterns. The combination of attribute-based scented widgets and selection-based relevance metrics enables experts to search through large attribute collections and refine classification results in multiple dimensions. Since the methodology exhibits the structure of time-dependent multivariate data, it is general and flexible enough to be applied in other domains.



Eventpad: Multivariate Collective Anomaly Exploration In Chapter 5 we focused on the design of a novel approach for analysts to explore multivariate event sequence data by combining attribute and sequential analysis into one unified system. The ability to interactively encode event logs by coalescing event sequences according to rules enables analysts to incorporate their knowledge of the data and test whether the patterns match their expectations.



The combination of attribute-based scented widgets and pattern aggregations enables ana-

lysts to discover new attributes of interest and refine rules based on their new findings, while staying aware of high-level patterns across different levels of abstractions. We have shown the effectiveness of the approach on real-world data sets through interactive sessions with external companies and elaborate examples. Since the methodology makes no underlying assumptions on sequential data, it is general and flexible enough to be used in other domains.

Hypothesis Testing & Generation in Wildlife traffic Chapter 6 presented a case study where we have shown the effectiveness of Eventpad to quickly gain insight in the VAST 2017



Mini Challenge 1 data set. We started exploration with the construction of hypotheses to reason about desired and undesired behavior after which they were verified during exploration of the data.

Using rules, aggregations, and selections, we discovered that vehicles on certain roads drive too fast and enter locations in the middle of the night for which they are not authorized. In addition, the presence of systematic travel activity across the entire preserve during high-season can also disturb the wildlife in Lekagul.

Rapid Reverse engineering of Malware Behavior In Chapter 7 we combined temporal analysis techniques from Chapter 4 with the Eventpad system to enable rapid and cost-effective



analysis of network traffic analysis and malware activity. We have shown the effectiveness of the system in real-world ransomware and office traffic. Our tool shows how visualization can be used to quickly gain insight in malware and network activity by combining data reduction and automated techniques in one highly interactive interface using rules, aggregations, and selections.

8.2. Reflections

Over the years we have demonstrated the systems at different companies (e.g., KPN, ASML, Philips, Motto communications) and security conferences including VizSec [322], Still Hacking Anyway [273], and Black Hat USA [29]. Application of the systems in practice and interaction with network engineers gave us valuable insights with respect to the design of cybersecurity visualizations. In this section we summarize our findings and reflect on the lessons learned.

8.2.1. System components & Integration

In practice there is no security technique that is able to detect all possible attacks. Every tool has its pros and cons and this is something that we have to accept. To this end we do not propose yet another unified framework for the exploration for security data or event logs, but focus on the visual components and paradigms that turned out to be most effective in each prototype (also illustrated in Figure 8.1).

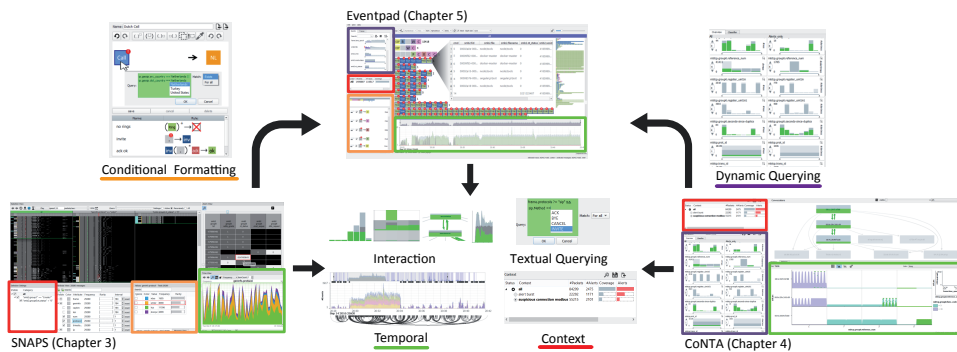


Figure 8.1: An overview of the components that were considered valuable in security visual analytics prototypes of Chapters 3, 4, and 5.

- *Context*: Although all prototypes were designed to detect a specific class of anomalies, some techniques have shown to be also relevant for the detection of other classes. In SNAPS we discovered that the application of a classifier on subsets of the data can greatly influence the resulting outcome. The Context view (red square, Figure 8.1) enabled users to quickly inspect data subsets by storing selections of interests throughout exploration. Combined with the classifiers in SNAPS and CoNTA, and the data operations in Eventpad, users were enabled to quickly analyze the data from different perspectives. In Chapter 2 context has shown to be important in the detection of anomalies and therefore played a crucial role in all three prototypes.
- *Conditional formatting*: Conditional formatting refers to the visual encoding of values and data collections according to user-defined conditions of interest. The SNAPS system enabled users to assign colors to values of interest and highlight these in the pixel visualization. In contrast to knowledge-based intrusion detection models, the rules enable users to discover new patterns in the visualization. This already showed an application where rules were not used to detect known attack patterns automatically, but to assist the user in discovering unknown patterns using visualization. In Eventpad conditional formatting became the main building block for the exploration and analysis of collective anomalies. Here we extended the technique from coloring values to the conditional coloring of sequences using regular expressions.
- *Dynamic Querying & Textual Querying*: Despite the preference of security analysts and Linux users to use Command-Line Interfaces (CLIs) over Graphical User Interfaces (GUIs) [136, 143], dynamic querying using sliders and scented widgets [333] was well received during demonstration of the systems. The creation of selections of interests in CoNTA and Eventpad enabled faster inspection of overlap in multiple attributes compared to command-line based queries. For the filtering of data with Boolean expressions, a traditional textual interface seemed more appropriate. In all prototypes dynamic querying and textual querying were provided to obtain the best from both worlds.
- *Temporal analysis*: In network monitoring, the detection of bursts and drops in network activity is often used as an indicator to determine the severity of changes in the

network. Many security threats such as Distributed Denial Of Service attacks (DDOS [220]), ransomware [219], DNS poisoning [308], but also the malfunction of hardware components [60] can be related to bursts and drops in network values over time. Similar to systems such as Grafana [123], VoipMonitor [282], and Kibana [127], we also believe that line charts are still valuable to study trends in this type of data.

- *Interaction*: In all systems *interaction* plays an important role as the amount of data is huge and the interest of the user has to be taken into account. Direct manipulation on graphical elements such as blocks, sliders, and diagrams enables users to instantly query data and obtain new insights. Combined with the creation of selections of interest and linking & brushing, nontrivial relationships can be discovered across multiple data views.

SNAPS enables direct manipulation by creating selections of interest in the pixel visualization and inspecting these using an interactive lens. Similar to Van der Corput et al. [318], Van den Elzen et al. [314], and many other modern systems for information visualization, CoNTA uses small multiples and scented widgets to enable the analysis of multivariate data in large collections of attributes. The multivariate regular expressions in Eventpad enable users to search, highlight, and define new patterns using a visual query interface.

We have also tried to use interaction to steer automated techniques in different ways. In SNAPS and CoNTA we retrained the classifier by interactively excluding values, ranges, or whole attributes from the data set. However, in practice we noticed that the effect on the classifier after applying on subsets of the data was sometimes hard to predict. In Eventpad we did not use a classification technique, but tried to steer automated techniques by defining similarity functions according to human labeling.

The development of SNAPS and CoNTA lead to valuable insights for the design of Eventpad. Although each system was designed to cover a particular class of anomaly (as illustrated in Chapter 1 Figure 1.3), we don't believe that the systems together should be considered as the full end solution to the research question. The introduction of user-centered rule generation in Eventpad opens a new perspective on how to analyze anomalies in event logs. As a consequence, we believe that the SNAPS and CoNTA prototype can also benefit from the techniques that were designed for Eventpad. Possibilities for future work are discussed in Section 8.3.

8.2.2. Eventpad: Interpretability in Security Visualization?

There are possible explanations for the success of Eventpad. We believe that they are related to the expressiveness, interpretability, and general character of the system. The visual one-to-one mapping of log records to blocks is relatively easy to comprehend and enables user to reason about every record individually without having to think about predefined aggregations or complex visual encodings. In addition, the visual encoding is widely applicable as it does not make any assumptions about the multivariate data attached to the records.

The data operations defined in Chapter 2 Figure 2.14 enable users to apply conceptually simple yet powerful transformations to data collections. In contrast to complex black-box processing in general, these operations are atomic, comprehensible, chainable, and reversible. Every operation is manually verifiable and again results in the same visual representation. We believe that the operations give the user control and help them assist in verifying the importance of observations based on the steps they took throughout exploration. The idea of systematically rewriting data using rules also fits in this paradigm.

The introduction of regular expressions to support the data operations such as grouping, deletion, and insertion of data points has already proven to be useful in the text editing domain. Although patterns as defined by regular expressions are rather basic (e.g., do not incorporate time), they are well known in the computer science community and relatively easy to reason about conceptually. The ability to capture knowledge in new concepts and to re-use them in other exploration tasks enables users to communicate results to other team members.

We believe that the positive reception of Eventpad is closely related to the popularity of pivot tables in Excel. Pivot tables enable rapid analysis (e.g., sorting, counting, grouping) of tabular records without having to setup anything. It is quick, understandable, and capable to work with (almost) any type of structured data. In addition, functionality to assist in the conversion to CSV data and import/export results to other systems such as Wireshark enables people from other domains to use the software quicker.

8.2.3. A hybrid model for Network Intrusion Detection

In Chapter 2 we discussed different intrusion detection models for the detection of threats in cybersecurity. In general we know that

8

- Knowledge-based detection in general is fast in pattern recognition, but ineffective for the detection of unknown attacks (e.g., APTs and zero-days);
- Behavioral-based anomaly detection techniques are able to discover unknown attacks, but are sensitive to high false positive rates due to model overfitting [293];
- Specification-based techniques enable both the discovery of known and unknown attacks according to some expectancy model. The construction of a full system specification in practice however is often tedious and difficult.

Each technique has its pros and cons, we found that in practice they are all indispensable and should be used together. For the detection of APTs, the SNAPS and CoNTA systems followed a behavioral-based anomaly detection approach, since the visualizations and classification models use historical records to discover deviations in the network according to some baseline (instead of relying on known attack patterns). The Eventpad system however is more difficult to classify in one of these categories.

The regular expressions in Eventpad enable users to look for known patterns of interest similar to knowledge-based systems such as Snort [21]. However, the rules also enable users to visually encode properties that are of interest for the discovery of unknown patterns. Similar

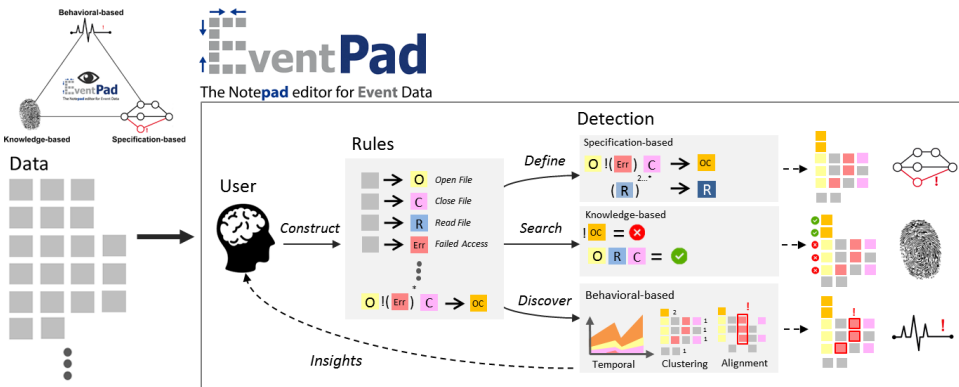


Figure 8.2: Eventpad is an example of a system where visualization and interaction are used to combine the strengths of multiple network intrusion detection approaches. Eventpad enables analysts to simplify event sequences by incrementally replacing block sequences using rules. Comparison between execution sequences is achieved by applying clustering and alignment techniques to the rewritten data. Inspection of attributes in overlapping sequences enables analysts to identify (un)desired behavior in the traffic. New insights can directly be incorporated in the analysis by defining new rules throughout exploration.

to behavioral-based intrusion detection, the severity of observed patterns can be assessed by comparing these to frequent or baseline behavior as suggested by clustering and alignment techniques. The incremental rewriting of event patterns enables users to visually separate desired and undesired sequences in the visualization according to their model of expectation. Even when analysts only have a partial specification of what desired behavior represents, interactive visualization enables users to incrementally obtain a better understanding of the data. In contrast to pure specification-based intrusion detection techniques, full system specifications are not required to gain insights. We believe that the success of Eventpad is related to its flexibility as the system does not try follow one these models, but aims to combine them (also illustrated in Figure 8.2).

The system shows that the role of visualization should not only be considered as a new technique to discover intrusions, but also as a means to bridge the gap between existing intrusion detection methodologies.

8.2.4. Automatic support: overview first, explanations on demand?

When we first applied the SNAPS system to real-world network traffic, we had initially no knowledge about good and bad behavior. When the first pixels started to light up, it was difficult to assess the severity of an alert by looking at its details. Inspection of regular traffic was necessary to put the alert in context. During the design of our prototypes we started to use automatic methods in different ways to explain occurrences of events.

The SNAPS and CoNTA systems in Chapters 3 and 4 followed an *algorithm-centered* approach where we considered the output of automatic methods as an input source for the visualization. In this top-down approach, we started with an overview of generated alerts while trying to find explanations of observed patterns in these collections. In SNAPS we

followed a more traditional security visualization approach where we generated an overview by statically visually encoding the alert data in one image, whereas in CoNTA we decided to alter the type of visualization depending on the task of interest. Although we were able to find explanations in the alerts by comparing them to normal traffic, sometimes it was still unclear why the classifier generated the alerts in the first place despite the simplicity of the model.

With Eventpad we took a different approach, where we enabled bottom-up discovery of anomalies by starting with user-defined patterns of interest. In this *user-centered* approach, analysts are enabled to find undesired or unexpected behavior by searching for deviations in patterns they expect to see in the network. Here automated techniques are used to assist the user in discovering new patterns and anomalies by starting from the context he defines.

Both algorithm-centered and user-centered approaches have advantages and disadvantages. Algorithm-centered support in general is easier to use in practice as it requires little domain knowledge from the user to discover anomalies and areas of interest. Assessing the relevance of an alert, however, can become difficult as it may require to find an explanation for the outcome of the underlying classification model.

Human-centered support assumes that the end user has knowledge about desired and undesired behavior. Automatic techniques in turn can use this knowledge to provide better context-sensitive insights in the area of interest. Although this level of awareness can be assumed for security analysts in Computer Emergency Response Teams and Security Operations Centers, users without any background knowledge may suffer from a cold-start problem. In addition, human-centered exploration can end in a local optimum as the search space of all possible contexts to inspect the data is biased towards the interests of the user. In the end we believe that both approaches can lead to valuable insights in the data and therefore should both be considered for the analysis of event collections.

8

8.2.5. Security starts with understanding

Network protocols nowadays are very complex. The RFC manual of a protocol such as SIP [266] has over 250 pages out of which at least 30% are describing exceptional cases. Over the years functionality has been added to and removed from protocol specifications that writing dissectors for them is tedious and error-prone. Ambiguities or unspecified behavior in specifications can be the source of a new vulnerability that can be exploited by targeted attacks. Similar to Etalle et al. [92], we believe that the high false positive rates of intrusion detection systems are more related to the complexity of network traffic rather than the quality of the detection techniques.

In principle, it is always possible to infiltrate a system, provided that the attacker has sufficient resources in terms of hardware, funding, time, and knowledge. The only thing that we can do is trying to raise the bar to make it less attractive for the attacker to infiltrate the environment. This also holds for the detection of targeted attacks. Although it is impossible to guarantee a 100% protection, we can increase the difficulty of being infiltrated by developing new techniques to inspect, assess, and most importantly trying to simplify our network traf-

fic. Similar to software standards, we believe that protocols should be indivisible, concise, and not subject to changes over time. API changes should only be accepted if the probability of someone exploiting the incompatibility as a result of the change is zero. Tulach refers to this paradigm as the “99 percent backward compatible API” [310].

8.2.6. Recommendations for building security tools

In this dissertation we have had successes and failures with respect to the deployment of our systems in practice. The Eventpad system was well received by companies and communities, whereas the SNAPS and CoNTA were less successful. We believe that in order for a security visualization tool to become of interest in practice, the following aspects need to be taken into account:

- *Integration & Transparency:* Besides discovering threats, cybersecurity teams also have the responsibility to justify decisions whenever actions have to be made. Especially with GDPR regulations [30], this is becoming more and more important. Over the years these teams have built their own complex platforms by combining (often open-source) frameworks in one environment. They are not interested in yet another black-box platform that will solve all their problems. Tools need to be integrateable and results need to be explainable. The SNAPS and CoNTA systems were already considered monolithic in the sense that import and export functionality was rather limited and alerts were sometimes still difficult to grasp. The Eventpad system provided better import and export functionality of rules and data to more universal formats and standards such as Comma Separated Values. In addition, the steps in the visualization after applying data operations such as clustering and alignment were easier to comprehend and verify compared to SNAPS and CoNTA.
- *Terminal support* The security community in general is rather skeptical on the use of graphical user interfaces [285, 297]. Command-Line Interfaces (CLI) enable:
 - faster execution since one does not have to look for or navigate to buttons on a screen;
 - easier repetition and history tracking, since CLIs store records of the executed commands;
 - fast input/output chaining (In Unix also referred to as “piping”);
 - scripting/automation of user tasks such as clicking a sequence of buttons.

In order to overcome limitations of graphical user interfaces, security tools should provide command-line functionality or provide fast interaction mechanism to support these features. In all three prototypes we enabled users to filter and export selections using a query interface similar to Wireshark [63]. Import and export functionality of rules in Eventpad enabled analysts to construct rules outside of the prototype. In addition, compatibility with open standards of STIX, TAXII and CyBOX [320] can also increase the utility of the tool in practice.

- *Expressiveness over flexibility:* In Chapter 2 we saw that security visualizations in the

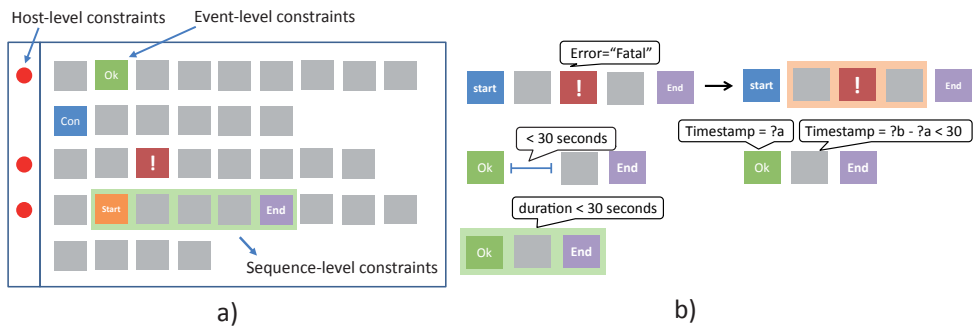


Figure 8.3: a) Eventpad can be extended with temporal constraints by defining constraints at the level of hosts, sequences, and events. b) Possible extension of the query language to support temporal constraints. The top-most query enforces conditional formatting over sequences of events. The other queries are three examples to enforce the time between events to be less than 30 seconds. $?a$ and $?b$ are unbound variables whose values are evaluated during the matching.

beginning were often focused towards a certain task or question, like: “show me deviations in port activity” or “show me where all alerts are located in the network”. As a result the main view of the visualization is often fixed according to the predefined question of interest. In Chapter 4 however we saw that the workflow of digital forensics is not limited to a single question.

We believe that a tool needs to support all basic data operations such as grouping, sorting, and filtering as defined in Chapter 2 in an effective and efficient way. This enables users to express their goal, quickly drill down, and verify their hypotheses on different subsets of the data without preprocessing the data first. Although SIEM dashboards offer a wide range of tools to visualize event collections, they often provide little functionality to manipulate the data interactively, making them unsuitable for in-depth data exploration.

8

8.3. Future work

Cybersecurity is a dynamic field where new techniques have to be developed continuously in order to counteract future cyberthreats. Although we have discovered novel ways to interact with network traffic and apply automated methods, there is still much to be done. Based on the feedback we received from industry and the suggestions for future work in the previous chapters, we believe there are several directions for future work with respect to the temporal analysis, comparison, and interaction of multivariate event log visualizations.

- *Temporal analysis:* The Eventpad system enables users to discover sequential patterns in event collections by ignoring the time between events. In both the security and health-care domain however, the absolute time of occurrence and relative time between events can tell much about the type of attack or the patient’s symptoms as a result of bad medication [263]. Although Eventpad has been extended to enable temporal analysis of rule invocations, this only solves the problem partly. Similar to EventFlow, users also want to

be enabled to impose timing constraints in the query language [223]. Figure 8.3 shows how the query language can be visually extended to support the specification of the constraints. The extension of the underlying regular expression model however is nontrivial as it may require backtracking to support the validation of temporal constraints.

- *Event log comparison:* In the SNAPS prototype we discovered that critical infrastructures showed repetitive behavior in event collections over time. An interesting topic for future work would be to analyze how snapshots of event collections change over time when recording the traffic at fixed intervals. Similar to Van den Elzen et al. [313], one idea would be to reduce event collections to two-dimensional points in space using dimensionality reduction techniques [106] and study evolutionary patterns in the event logs as a result of an attack.
- *Exploration of text-oriented and process-mining algorithms:* The visualization approach in Eventpad enables us to apply text-oriented techniques such as regular expression searching to multivariate event logs. In addition, process-mining also provides a scala of algorithms [176, 315, 317], besides multiple sequence alignment, to discover patterns in multivariate event sequences automatically. Although such techniques in general are computationally intensive, quick results on data selections can be obtained through user interaction. The text analysis domain also offers a wide variety of algorithms that could be used in Eventpad. Interesting future work would be to see how we can for instance compare event collections using `textdiff` [334] algorithms that are used in version control systems such as Git [306], or other text searching algorithms such as `tf-idf` [259] when defining words interactively using the Eventpad rules.
- *Heterogeneous data analysis:* In this dissertation we mainly focused on the analysis of event logs and implicitly assumed that the domain knowledge of the network analyst can provide enough context about the environment. In practice, however, this user experience is often obtained by analyzing different sources of information. Tweets on the web, news reports, camera surveillance videos, and sometimes even weather forecastings are analyzed to find explanations in computer networks. Although a wide variety of techniques have been proposed to visualize data sources individually, the development of a consistent and concise interaction mechanism between these views is challenging [269].
- *Multi-user interaction:* Incident response teams often have to work closely together when trying to resolve a system breach. Communicating results in these hectic periods however is difficult. There is a need for Computer Emergency Response Teams (CERTs) and Security Operations Centers (SOCs) to be able to communicate their findings when working on the same topic. This raises the question on how to enable multiple users to interact with the same visualization simultaneously.

8.4. In Conclusion

In this dissertation we have presented different techniques to combine visualization techniques and automated methods for the discovery of patterns and anomalies in large event collections. We have shown that visual analytics enables the discovery of application-level attacks by combining interaction, visualization, and automated methods in coherent systems. The SNAPS and CoNTA systems have shown that visualization can be effective to discover patterns in automatically generated alert collections, whereas systems such as Eventpad enables the discovery of user-driven patterns. We believe that visualization plays an important role in the integration of different intrusion detection techniques and raising awareness in the network. However, we also believe that the use of visualization only solves the detection of targeted attacks partly. To better understand what is happening in computer networks, we need to find ways to make networks and protocols less complex and better administrable. Security through obscurity [141] is never a complete solution, security starts with understanding, and we hope to have contributed to that with our work.

References

- [1] **Abdelnur, H., Avanesov, T., Rusinowitch, M., and State, R.** Abusing SIP Authentication. In *Proceedings of the International Conference on Information Assurance and Security (ISIAS)* (2008), pp. 237–242.
- [2] **Abdullah, K., Lee, C., Conti, G., and Copeland, J. A.** Visualizing Network Data for Intrusion Detection. In *Proceedings of the IEEE SMC Information Assurance Workshop (IAW)* (2005), pp. 100–108.
- [3] **Abdullah, K., Lee, C., Conti, G., Copeland, J. A., and Stasko, J.** IDS Rainstorm: Visualizing IDS Alarms. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 1–8.
- [4] **Aho, A., Sethi, R., and Ullman, J.** *Compilers, Principles, Techniques*. 1986.
- [5] **Aigner, W., Miksch, S., Thurnher, B., and Biffl, S.** Planninglines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proceedings of the International Conference on Information Visualisation* (2005), pp. 457–463.
- [6] **Akritidis, P., Markatos, E. P., Polychronakis, M., and Anagnostakis, K.** Stride: Polymorphic Sled Detection through Instruction Sequence Analysis. In *International Information Security Conference (IFIP)* (2005), pp. 375–391.
- [7] **AlienVault.** Alienvault Security Information and Event Management. <https://www.alienvault.com/>. Last Accessed: 24-05-2018.
- [8] **Anderson, B., Paul, S., and McGrew, D.** Deciphering Malware’s use of TLS (without Decryption). *Journal of Computer Virology and Hacking Techniques* (2016), 1–17.
- [9] **Anderson, B., Storlie, C., and Lane, T.** Improving Malware Classification: Bridging the Static/Dynamic Gap. In *Proceedings of the ACM Workshop on Security and Artificial Intelligence (AISec)* (2012), AISec ’12, pp. 3–14.
- [10] **Anderson, N.** Deep Packet Inspection under assault over Privacy Concerns. <http://arstechnica.com/news.ars/post/20080512-deep-packet-inspection-under-assault-from-canadian-critics.html>, 2008. Last Accessed: 30-05-2018.
- [11] **André, P., Wilson, M. L., Russell, A., Smith, D. A., Owens, A., et al.** Continuum: Designing Timelines for Hierarchies, Relationships and Scale. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (2007), pp. 101–110.
- [12] **Angelini, M., Aniello, L., Lenti, S., Santucci, G., and Ucci, D.** The Goods, the Bads and the Uglys: Supporting Decisions in Malware Detection through Visual Analytics. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2017), pp. 1–8.

- [13] **Angelini, M., Lenti, S., and Santucci, G.** CRUMBS: A Cyber Security Framework Browser. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2017), pp. 1–8.
- [14] **Angelini, M., Prigent, N., and Santucci, G.** Percival: Proactive and Reactive Attack and Response Assessment for Cyber Incidents using Visual Analytics. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–8.
- [15] **Arendt, D. L., Best, D., Burtner, R., and Paul, C. L.** Cyberpetri at CDX 2016: Real-time Network Situation Awareness. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), pp. 1–4.
- [16] **Arendt, D. L., Burtner, R., Best, D. M., Bos, N. D., Gersh, J. R., Piatko, C. D., and Paul, C. L.** Ocelot: User-centered Design of a Decision Support Visualization for Network Quarantine. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–8.
- [17] **Attipoe, A. E., Yan, J., Turner, C., and Richards, D.** Visualization Tools for Network Security. *Electronic Imaging* 2016, 1 (2016), 1–8.
- [18] **Avallone, J.** Regexper. <https://regexper.com/>, 2010. Last Accessed: 01-07-2018.
- [19] **Axelsson, S., and Sands, D.** Combining a Bayesian classifier with Visualization: Understanding the IDS. *Understanding Intrusion Detection Through Visualization* (2006), 69–87.
- [20] **Ball, R., Fink, G. A., and North, C.** Home-centric Visualization of Network Traffic for Security Administration. In *Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security* (2004), pp. 55–64.
- [21] **Beale, J., Baker, A., Esler, J., and Kohlenberg, T.** *Snort: IDS and IPS Toolkit*. Syngress, 2007.
- [22] **Beaugnon, A., Chifflier, P., and Bach, F.** ILAB: An Interactive Labelling Strategy for Intrusion Detection. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)* (2017), pp. 120–140.
- [23] **Bendrath, R., and Mueller, M.** The End of The Net as we know it? Deep Packet Inspection and Internet Governance. *New Media & Society* 13, 7 (2011), 1142–1160.
- [24] **Bernard, J., Sessler, D., May, T., Schlomm, T., Pehrke, D., and Kohlhammer, J.** A Visual-interactive System for Prostate Cancer Cohort Analysis. *IEEE Computer Graphics and Applications* 35, 3 (2015), 44–55.
- [25] **Bernard, J., Zeppelzauer, M., Sedlmair, M., and Aigner, W.** A Unified Process for Visual-Interactive Labeling. *Proceedings of the EuroVis Workshop on Visual Analytics* (2017).

- [26] **Bertini, E., Hertzog, P., and Lalanne, D.** Spiralview: Towards Security Policies Assessment through Visual Correlation of Network Resources with Evolution of Alarms. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)* (2007), pp. 139–146.
- [27] **Black, P. E.** Manhattan distance. *Dictionary of Algorithms and Data Structures*. 18 (2006), 2012.
- [28] **Black, U.** *Voice over IP*. Prentice-Hall, Inc., 1999.
- [29] **Black Hat.** Black hat. <https://www.blackhat.com/us-18/>, 2018. Last Accessed: 04-09-2018.
- [30] **Blackmer, W.** GDPR: Getting Ready for the New EU General Data Protection Regulation. *Information Law Group* 22, 08 (2016), 2016.
- [31] **Bose, R., and der Aalst, W. V.** Trace Alignment in Process Mining: Opportunities for Process Diagnostics. In *Proceedings of the International Conference on Business Process Management* (2010), pp. 227–242.
- [32] **Bratbergsengen, K.** Hashing methods and Relational Algebra Operations. In *Proceedings of the International Conference on Very Large Data Bases* (1984), pp. 323–333.
- [33] **Brehmer, M., and Munzner, T.** A Multi-level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2376–2385.
- [34] **Briscoe, N.** Understanding the OSI 7-layer Model. *PC Network Advisor* 120, 2 (2000).
- [35] **Brown, A.** I3: Maximizing Packet Capture Performance. *Wireshark Developer and User Conference (Sharkfestus)* (2014). Last Accessed: 30-05-2018.
- [36] **Bugzilla.** Microsoft Samba Bugzilla. https://bugzilla.samba.org/show_bug.cgi?id=7331. Last Accessed: 30-05-2018.
- [37] **Bugzilla.** Microsoft Samba Bugzilla. https://bugzilla.samba.org/show_bug.cgi?id=9173. Last Accessed: 30-05-2018.
- [38] **Bugzilla.** Microsoft Samba Bugzilla. https://bugzilla.samba.org/show_bug.cgi?id=8407. Last Accessed: 30-05-2018.
- [39] **Buja, A., McDonald, J. A., Michalak, J., and Stuetzle, W.** Interactive Data Visualization using Focusing and Linking. In *Proceedings of the IEEE Conference on Visualization* (1991), pp. 156–163.
- [40] **Burch, M., Beck, F., and Diehl, S.** Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies. In *Proceedings of the working conference on Advanced Visual Interfaces* (2008), pp. 75–82.

- [41] **Cai, Y., and de M. Franco, R.** Interactive Visualization of Network Anomalous Events. In *Proceedings of the International Conference on Computational Science* (2009), pp. 450–459.
- [42] **Camiña, J., Rodríguez, J., and Monroy, R.** Towards a Masquerade Detection System based on User's Tasks. In *Proceedings of the International Workshop on Recent Advances in Intrusion Detection (RAID)* (2014), pp. 447–465.
- [43] **Cappers, B. C. M.** Exploring Lekagul Sensor Events using Rules, Aggregations, and Selections. *Proceedings of the IEEE Visual Analytics Science and Technology Challenge (VAST)* (2017).
- [44] **Cappers, B. C. M., Meessen, P. N., Etalle, S., and van Wijk, J. J.** Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics. *Expected in Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2018).
- [45] **Cappers, B. C. M., and van Wijk, J. J.** SNAPS: Semantic Network Traffic Analysis through Projection and Selection. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–8.
- [46] **Cappers, B. C. M., and van Wijk, J. J.** Understanding the Context of Network Traffic Alerts. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), pp. 1–8.
- [47] **Cappers, B. C. M., and van Wijk, J. J.** Exploring Multivariate Event Sequences using Rules, Aggregations, and Selections. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 532–541.
- [48] **Card, S. K., Mackinlay, J. D., and Shneiderman, B.** Using Vision to Think. In *Readings in Information Visualization* (1999), pp. 579–581.
- [49] **Carson, T. L.** Conflicts of Interest. *Journal of Business Ethics* 13, 5 (1994), 387–404.
- [50] **Casey, E.** *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic press, 2011.
- [51] **Chandola, V., Banerjee, A., and Kumar, V.** Anomaly Detection: A Survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 15:1–15:58.
- [52] **Chapelle, O., Scholkopf, B., and Zien, A.** Semi-supervised Learning. *IEEE Transactions on Neural Networks* 20, 3 (2009), 542–542.
- [53] **Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R.** CRISP-DM 1.0 Step-By-Step Data Mining Guide. *Technical Report of the CRISP-DM Consortium* (2000).
- [54] **Chen, H., Chiang, R. H., and Storey, V. C.** Business Intelligence and Analytics: from Big Data to Big Impact. *MIS quarterly* (2012), 1165–1188.

- [55] **Chen, S., Chen, S., Lin, L., Yuan, X., Liang, J., and Zhang, X.** E-map: A Visual Analytics approach for Exploring Significant Event Evolutions in Social Media. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017).
- [56] **Chen, S., Guo, C., Yuan, X., Merkle, F., Schaefer, H., and Ertl, T.** Oceans: Online Collaborative Explorative Analysis on Network Security. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2014), pp. 1–8.
- [57] **Chen, Y., Xu, P., and Ren, L.** Sequence Synopsis: Optimize Visual Summary of Temporal Event Data. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 45–55.
- [58] **Cho, I., Wesslen, R., Volkova, S., Ribarsky, W., and Dou, W.** Crystalball: A Visual Analytic System for Future Event Discovery and Analysis from Social Media Data. *IEEE Transactions on Visualization and Computer Graphics* (2018).
- [59] **Claise, B.** Cisco Systems Netflow Services. <https://tools.ietf.org/html/rfc3954>, 2004. Last Accessed: 01-07-2018.
- [60] **Clayton, M.** Stuxnet malware is “weapon” out to destroy... Iran’s Bushehr Nuclear Plant. *Christian Science Monitor* 21 (2010).
- [61] **Colitti, L., Di Battista, G., Mariani, F., Patrignani, M., and Pizzonia, M.** Visualizing Interdomain Routing with BGPlay. *Journal on Graph Algorithms Applications* 9, 1 (2005), 117–148.
- [62] **Colombe, J. B., and Stephens, G.** Statistical Profiling and Visualization for Detection of Malicious Insider Attacks on Computer Networks. In *Proceedings of the ACM workshop on Visualization and Data Mining for Computer Security* (2004), pp. 138–142.
- [63] **Combs, G., et al.** Wireshark Network Protocol Analyzer. <http://www.wireshark.org/>, 2008. Last Accessed: 30-05-2018.
- [64] **Conti, G., Abdullah, K., Grizzard, J., Stasko, J., Copeland, J. A., Ahamad, M., Owen, H. L., and Lee, C.** Countering Security Information Overload through Alert and Packet Visualization. *IEEE Computer Graphics and Applications* 26, 2 (2006), 60–70.
- [65] **Conti, G., and Dean, E.** Visual Forensic Analysis and Reverse Engineering of Binary Data. *Black Hat USA* (2008).
- [66] **Conti, G., Grizzard, J., Ahamad, M., and Owen, H.** Visual Exploration of Malicious Network Objects using Semantic Zoom, Interactive Encoding and Dynamic Queries. In *Proceedings of the IEEE Worksop on Visualization for Cyber Security (VizSec)* (2005), pp. 83–90.

- [67] **Cook, K., Grinstein, G., and Whiting, M.** VAST Challenge 2017: Mystery at the Wildlife Preserve. *Proceedings of the Visual Analytics Science and Technology (VAST) Challenge* (2017).
- [68] **Cook, K., Grinstein, G., and Whiting, M.** Visual Analytics Science and Technology (VAST) Challenge. <http://www.vacommunity.org/VAST+Challenge+2017/>, 2017. Last Accessed: 03-07-2018.
- [69] **Cortese, P. F., Di Battista, G., Moneta, A., Patrignani, M., and Pizzonia, M.** Topographic Visualization of Prefix Propagation in the Internet. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 725–732.
- [70] **Costante, E., den Hartog, J., Petković, M., Etalle, S., and Pechenizkiy, M.** Hunting the Unknown. In *IFIP Annual Conference on Data and Applications Security and Privacy* (2014), pp. 243–259.
- [71] **Cowan, C., Pu, C., Maier, D., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., Zhang, Q., and Hinton, H.** Stackguard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. In *Usenix Security* (1998), vol. 98, pp. 63–78.
- [72] **D’Amico, A., and Whitley, K.** The Real Work of Computer Network Defense Analysts. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2008), pp. 19–37.
- [73] **Danowski, J. A.** Network Analysis of Message Content. *Progress in Communication Sciences* 12 (1993), 198–221.
- [74] **Dasgupta, A., Arendt, D. L., Franklin, L. R., Wong, P. C., and Cook, K. A.** Human Factors in Streaming Data Analysis: Challenges and Opportunities for Information Visualization. In *Computer Graphics Forum* (2018), vol. 37, pp. 254–272.
- [75] **Dashora, K.** Cyber Crime in the Society: Problems and Preventions. *Journal of Alternative Perspectives in the Social Sciences* 3, 1 (2011), 240–259.
- [76] **Deza, M. M., and Deza, E.** Encyclopedia of Distances. In *Encyclopedia of Distances* (2009), pp. 1–583.
- [77] **Dhar, V.** Data Science and Prediction. *Communications of the ACM* 56, 12 (2013), 64–73.
- [78] **Dierks, T.** The Transport Layer Security (TLS) Protocol version 1.2. <https://tools.ietf.org/pdf/rfc5246>, 2008. Last Accessed: 30-05-2018.
- [79] **Domas, C.** Cantordust. <https://sites.google.com/site/xxcantorxdustxx/>, 2012. Last Accessed: 30-05-2018.
- [80] **Donahue, J., Paturi, A., and Mukkamala, S.** Visualization Techniques for Efficient Malware Detection. In *Proceedings of the IEEE Conference on Intelligence and Security Informatics (ISI)* (2013), pp. 289–291.

- [81] **Dreyfus, S.** The Security Paradox Individual Privacy versus Digital Driftnets. <http://theconversation.com/the-security-paradox-individual-privacy-versus-digital-driftnets-38910>, 2015. Last Accessed: 30-05-2018.
- [82] **Du, F., Shneiderman, B., Plaisant, C., Malik, S., and Perer, A.** Coping with Volume and Variety in Temporal Event Sequences: Strategies for Sharpening Analytic Focus. *IEEE Transactions on Visualization and Computer Graphics* (2016).
- [83] **Dua, S., and Du, X.** *Data Mining and Machine Learning in Cybersecurity*. CRC press, 2016.
- [84] **Eckerson, W. W.** Predictive Analytics: Extending the Value of Your Data Warehousing Investment. *TDWI Best Practices Report 1* (2007), 1–36.
- [85] **Eick, S. G., Nelson, M. C., and Schmidt, J. D.** Graphical Analysis of Computer Log Files. *Communications of the ACM* 37, 12 (1994), 50–56.
- [86] **Elvins, T. T.** A Survey of Algorithms for Volume Visualization. *ACM Siggraph Computer Graphics* 26, 3 (1992), 194–201.
- [87] **Engel, F., Jones, K. S., Robertson, K., Thompson, D. M., and White, G.** Network Monitoring, 2000. US Patent 6,115,393.
- [88] **Erbacher, R. F.** Intrusion Behavior Detection through Visualization. In *IEEE International Conference on Systems, Man and Cybernetics* (2003), vol. 3, pp. 2507–2513.
- [89] **Erbacher, R. F., Christensen, K., and Sundberg, A.** Designing Visualization Capabilities for IDS Challenges. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSec)* (2005), pp. 121–127.
- [90] **Erbacher, R. F., Walker, K. L., and Frincke, D. A.** Intrusion and Misuse Detection in Large-scale Systems. *IEEE Computer Graphics and Applications* 22, 1 (2002), 38–47.
- [91] **Ester, M., Kriegel, H.-P., Sander, J., and Xu, X.** Density-based Spatial Clustering of Applications with Noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining* (1996), vol. 240.
- [92] **Etalle, S.** From Intrusion Detection to Software Design. *Proceedings of the European Symposium on Research in Computer Security (ESORICS)* (2017), 1–10.
- [93] **Everitt, B., Landau, S., Leese, M., and Stahl, D.** Hierarchical Clustering. *Cluster Analysis, 5th Edition* (2011), 71–110.
- [94] **Fails, J. A., Karlson, A., Shahamat, L., and Shneiderman, B.** A Visual Interface for Multivariate Temporal Data: Finding patterns of Events across Multiple Histories. In *Proceedings of the IEEE Symposium on Visual Analytics Science And Technology* (2006), pp. 167–174.

- [95] **Fast Company**. Chinese Hackers Target New York Times Four Month Cyberattack. <https://www.fastcompany.com/3005280/>, 2015. Last Accessed: 30-05-2018.
- [96] **Fauri, D., de Wijs, B., den Hartog, J., Costante, E., Etalle, S., and Zambon, E.** Encryption in ICS Networks: a Blessing or a Curse. *Technical report, Eindhoven Technical University* (2017).
- [97] **Ferebee, D., and Dasgupta, D.** Security Visualization Survey. In *Proceedings of the Colloquium for Information Systems Security Education University of Texas* (2008), p. 124.
- [98] **Field, S. A.** Tagging obtained content for White and Black Listing, 2013. US Patent 8,544,086.
- [99] **Fink, G. A., Duggirala, V., Correa, R., and North, C.** Bridging the Host-Network Divide: Survey, Taxonomy, and Solution. In *LISA* (2006), pp. 247–262.
- [100] **Fink, G. A., Muessig, P., and North, C.** Visual Correlation of Host Processes and Network Traffic. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 11–19.
- [101] **FireEye**. Attackers deploy new ICS attack framework Triton. <https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html>, 2017. Last Accessed: 30-05-2018.
- [102] **First Post**. Cyber crime now 'number one' threat: Europol chief says it's a global problem. <https://www.firstpost.com/world/cyber-crime-now-number-one-threat-europol-chief-says-global-problem-2202120.html>, 2018. Last Accessed: 30-05-2018.
- [103] **Fischer, F., Fuchs, J., Vervier, P.-A., Mansmann, F., and Thonnard, O.** Vistracer: a Visual Analytics Tool to Investigate Routing Anomalies in Traceroutes. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2012), pp. 80–87.
- [104] **Fischer, F., and Keim, D. A.** NStreamAware: Real-time Visual Analytics for Data Streams to Enhance Situational Awareness. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2014), pp. 65–72.
- [105] **Fischer, F., Mansmann, F., Keim, D. A., Pietzko, S., and Waldvogel, M.** Large-scale Network Monitoring for Visual Analysis of Attacks. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2008), pp. 111–118.
- [106] **Fodor, I. K.** A Survey of Dimension Reduction Techniques. *Technical Report Lawrence Livermore National Lab.* (2002).
- [107] **Foresti, S., Agutter, J., Livnat, Y., Moon, S., and Erbacher, R.** Visual Correlation of Network Alerts. *IEEE Computer Graphics and Applications* 26, 2 (2006), 48–59.

- [108] **Forouzan, B. A., and Fegan, S. C.** *TCP/IP protocol suite*. McGraw-Hill Higher Education, 2002.
- [109] **Fowler, J. J., Johnson, T., Simonetto, P., Schneider, M., Acedo, C., Kobourov, S., and Lazos, L.** IMap: Visualizing Network Activity over Internet Maps. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2014), pp. 80–87.
- [110] **Fox, D.** Computer Emergency Response Team (CERT). *Datenschutz und Datensicherheit* 26, 8 (2002).
- [111] **Friedl, J. E.** *Mastering Regular Expressions*. O'Reilly Media, Inc., 2002.
- [112] **Fuchs, C.** Societal and Ideological Impacts of Deep Packet Inspection Internet Surveillance. *Information, Communication & Society* 16, 8 (2013), 1328–1359.
- [113] **Gansner, E., Koutsofios, E., North, S., et al.** A Technique for Drawing Directed Graphs. *IEEE Transactions on Software Engineering* 19, 3 (1993), 214–230.
- [114] **Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E.** Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges. *Journal on Computers & Security* 28, 1-2 (2009), 18–28.
- [115] **Garfinkel, S.** Digital forensics Research: The next 10 years. *Digital Investigation* 7 (2010), S64–S73.
- [116] **Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, S., Sisalem, D., et al.** Survey of Security Vulnerabilities in Session Initiation Protocol. *IEEE Communications Surveys and Tutorials* 8, 1-4 (2006), 68–81.
- [117] **Giacobe, N. A., and Xu, S.** Geovisual Analytics for Cyber Security: Adopting the GeoViz Toolkit. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011), pp. 315–316.
- [118] **Girardin, L.** An Eye on Network Intruder-Administrator Shootouts. In *Workshop on Intrusion Detection and Network Monitoring* (1999), pp. 19–28.
- [119] **Glanfield, J., Brooks, S., Taylor, T., Paterson, D., Smith, C., Gates, C., and McHugh, J.** Over Flow: An Overview Visualization for Network Analysis. In *Proceedings of the International Workshop on Visualization for Cyber Security (VizSec)* (2009), pp. 11–19.
- [120] **Goodall, J. R., Lutters, W. G., Rheingans, P., and Komlodi, A.** Preserving the Big Picture: Visual Network Traffic Analysis with TNV. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSec)* (2005), pp. 47–54.
- [121] **Gormley, C., and Tong, Z.** *Elasticsearch: The Definitive Guide*. ” O'Reilly Media, Inc.”, 2015.
- [122] **Gotz, D., and Stavropoulos, H.** Decisionflow: Visual Analytics for High-dimensional Temporal Event Sequence Data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1783–1792.

- [123] **Grafana Labs.** Grafana. <https://grafana.com/>, 2018. Last Accessed: 25-06-2018.
- [124] **Granger, S.** Social Engineering Fundamentals, Part I: Hacker Tactics. *Security Focus* 18 (2001).
- [125] **Guimaraes, V. T., Freitas, C. M. D. S., Sadre, R., Tarouco, L. M. R., and Granville, L. Z.** A Survey on Information Visualization for Network and Service Management. *IEEE Communications Surveys and Tutorials* 18, 1 (2016), 285–323.
- [126] **Guo, S., Xu, K., Zhao, R., Gotz, D., Zha, H., and Cao, N.** EventThread: Visual Summarization and Stage Analysis of Event Sequence Data. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 56–65.
- [127] **Gupta, Y.** *Kibana Essentials*. Packt Publishing Ltd, 2015.
- [128] **Gvily, Y.** User Impersonation by a Proxy Server, May 6 2008. US Patent 7,370,015.
- [129] **Hadžiosmanović, D., Bolzoni, D., Etalle, S., and Hartel, P.** Challenges and Opportunities in Securing Industrial Control Systems. In *Complexity in Engineering (COM-PENG)* (2012), pp. 1–6.
- [130] **Hadžiosmanović, D., Simionato, L., Bolzoni, D., Zambon, E., and Etalle, S.** N-gram Against the Machine: On the Feasibility of the N-gram Network Analysis for Binary Protocols. In *International Workshop on Recent Advances in Intrusion Detection (RAID)* (2012), pp. 354–373.
- [131] **Hájek, A.** The Reference Class Problem is your Problem too. *Synthese* 156, 3 (2007), 563–585.
- [132] **Hanna, A. R., Rao, C., and Athanasiou, T.** Graphs in Statistical Analysis. In *Key Topics in Surgical Research and Methodology* (2010), pp. 441–475.
- [133] **Hansman, S., and Hunt, R.** A Taxonomy of Network and Computer Attacks. *Computers & Security* 24, 1 (2005), 31–43.
- [134] **Hao, L., Healey, C. G., and Hutchinson, S. E.** Flexible Web Visualization for Alert-based Network Security Analytics. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2013), pp. 33–40.
- [135] **Hao, L., Healey, C. G., and Hutchinson, S. E.** Ensemble Visualization for Cyber Situation Awareness of Network Security Data. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–8.
- [136] **Harrington, J.** GUI versus CLI in Networking. <https://www.networkcomputing.com/networking/gui-vs-cli-networking/2083404544>, 2016. Last Accessed: 30-05-2018.
- [137] **Harrison, B. L., Owen, R., and Baecker, R. M.** Timelines: an Interactive System for the Collection and Visualization of Temporal Data. In *Graphics Interface* (1994), pp. 141–141.

- [138] **Hauser, H., Weiskopf, D., Ma, K.-L., van Wijk, J. J., and Kosara, R.** Scivis, Infovis Bridging the Community Divide. *IEEE Visualization Panel Proceedings* (2006), 52–55.
- [139] **Healey, C. G., Hao, L., and Hutchinson, S. E.** Lessons Learned: Visualizing Cyber Situation Awareness in a Network Security Domain. In *Theory and Models for Cyber Situation Awareness* (2017), pp. 47–65.
- [140] **Hickman, K., and Elgamal, T.** The SSL Protocol. *Netscape Communications Corp. 501* (1995).
- [141] **Hoepman, J.-H., and Jacobs, B.** Increased Security through Open Source. *Communications of the ACM* 50, 1 (2007), 79–83.
- [142] **Hoerbst, A., and Ammenwerth, E.** Electronic Health Records. *Methods of Information in Medicine* 49, 04 (2010), 320–336.
- [143] **Horne, B.** Why do so many Linux users prefer the command line to a GUI. <https://www.quora.com/Why-do-so-many-Linux-users-prefer-the-command-line-to-a-GUI>, 2017. Last Accessed: 30-05-2018.
- [144] **HP.** HP Arcsight Security Information and Event Management. <http://www8.hp.com/us/en/software-solutions/siem-security-information-event-management/>. Last Accessed: 24-05-2018.
- [145] **Huff, D.** *How to Lie with Statistics*. WW Norton & Company, 1993.
- [146] **Humphries, C., Prigent, N., Bidan, C., and Majorczyk, F.** Elvis: Extensible Log Visualization. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2013), pp. 9–16.
- [147] **Hutchins, E. M., Cloppert, M. J., and Amin, R. M.** Intelligence-driven Computer Network Defense informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Leading Issues in Information Warfare & Security Research* 1, 1 (2011), 80.
- [148] **Huynh, N. A., Ng, W. K., Ulmer, A., and Kohlhammer, J.** Uncovering Periodic Network Signals of Cyber Attacks. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), pp. 1–8.
- [149] **IBM.** IBM QRadar Security Information and Event Management. www.ibm.com/qradar. Last Accessed: 24-05-2018.
- [150] **Iliinsky, N., and Steele, J.** *Designing Data Visualizations: Representing Informational Relationships*. O'Reilly Media, Inc., 2011.
- [151] **Information Age.** Cybercrime: an unprecedented Threat to Society? <http://www.information-age.com/cyber-crime-greatest-threat-society-123464389/>, 2018. Last Accessed: 30-05-2018.

- [152] **Inoue, D., Eto, M., Suzuki, K., Suzuki, M., and Nakao, K.** DAEDALUS-VIZ: Novel Real-time 3D Visualization for Darknet Monitoring-based Alert System. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2012), pp. 72–79.
- [153] **Inoue, D., Eto, M., Yoshioka, K., Baba, S., Suzuki, K., Nakazato, J., Ohtaka, K., and Nakao, K.** Nictor: An Incident Analysis System toward Binding Network Monitoring with Malware Analysis. In *Proceedings of the Workshop on Information Security Threats Data Collection and Sharing (WISTDCS)* (2008), pp. 58–66.
- [154] **Israel, G. D.** *Determining sample size.* University of Florida Cooperative Extension Service, Institute of Food and Agriculture Sciences, EDIS, 1992.
- [155] **Jacobs, J., and Rudis, B.** *Data-driven security: analysis, visualization and dashboards.* John Wiley & Sons, 2014.
- [156] **Jain, A., Murty, N., and Flynn, P.** Data Clustering: a Review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.
- [157] **Janies, J.** Existence Plots: A Low-resolution Time series for Port behavior Analysis. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2008), pp. 161–168.
- [158] **Jen, D., Meisel, M., Massey, D., Wang, L., Zhang, B., and Zhang, L.** APT: A Practical Transit Mapping Service. *draft-jen-apt-01* (2007).
- [159] **Johansen, G.** *Digital Forensics and Incident Response.* Packt Publishing Ltd, 2017. Last Accessed: 30-05-2018.
- [160] **John, J. T.** State of the Art Analysis of Defense Techniques against Advanced Persistent Threats. *Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM) Focal Topic: Advanced Persistent Threats* 63 (2017).
- [161] **Karam, G. M.** Visualization using Timelines. In *Proceedings of the International Symposium on Software Testing and Analysis (SIGSOFT)* (1994), pp. 125–137.
- [162] **Kasemsri, R. R.** A Survey, Taxonomy, and Analysis of Network Security Visualization Techniques. *Thesis, Georgia State University* (2006).
- [163] **Kehrer, J., and Hauser, H.** Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (2013), 495–513.
- [164] **Keim, D.** Designing Pixel-oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics* 6, 1 (2000), 59–78.
- [165] **Keim, D. A.** Visual Exploration of Large Data Sets. *Communications of the ACM* 44, 8 (2001), 38–44.
- [166] **Keim, D. A.** Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8.

- [167] **Keim, D. A., Mansmann, F., Schneidewind, J., and Schreck, T.** Monitoring Network Traffic with Radial Traffic Analyzer. In *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology (VAST)* (2006), pp. 123–128.
- [168] **Kernighan, B. W., and Pike, R.** *The Unix Programming Environment*, vol. 270. Prentice-Hall Englewood Cliffs, NJ, 1984.
- [169] **Khanh Dang, T., and Tri Dang, T.** A Survey on Security Visualization Techniques for Web Information Systems. *International Journal of Web Information Systems* 9, 1 (2013), 6–31.
- [170] **Kim, H., Ko, S., Kim, D. S., and Kim, H. K.** Firewall Ruleset Visualization Analysis Tool based on Segmentation. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2017), pp. 1–8.
- [171] **Kittler, J., Hatef, M., Duin, R., and Matas, J.** On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 3 (1998), 226–239.
- [172] **Kleene, S.** Representation of Events in Nerve Nets and Finite Automata. *Technical Report, DTIC Document* (1951).
- [173] **Ko, C., Ruschitzka, M., and Levitt, K.** Execution Monitoring of Security-critical Programs in Distributed Systems: A Specification-based Approach. In *Proceedings of the IEEE Symposium on Security and Privacy (VizSec)* (1997), pp. 175–187.
- [174] **Koike, H., and Ohno, K.** Snortview: Visualization System of Snort Logs. In *Proceedings of the ACM workshop on Visualization and Data Mining for Computer Security* (2004), pp. 143–147.
- [175] **Koike, H., Ohno, K., and Koizumi, K.** Visualizing Cyber Attacks using IP Matrix. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 91–98.
- [176] **Kosala, R., and Blockeel, H.** Web Mining Research: A Survey. *ACM Sigkdd Explorations Newsletter* 2, 1 (2000), 1–15.
- [177] **Kosara, R., and Mackinlay, J.** Storytelling: The Next Step for Visualization. *IEEE Computer (Special Issue on Cutting-Edge Research in Visualization)* 46, 5 (2013), 44–50.
- [178] **Kott, A., Wang, C., and Erbacher, R.** *Cyber Defense and Situational Awareness*. 2014.
- [179] **KPN.** KPN publishes fifth European Cyber Security Perspectives. <https://overons.kpn/en/news/2018/kpn-publishes-fifth-european-cyber-security-perspectives>, 2018. Last Accessed: 01-07-2018.
- [180] **Krasser, S., Conti, G., Grizzard, J., Gribschaw, J., and Owen, H.** Real-time and Forensic Network Data Analysis using Animated and Coordinated Visualization. In *Proceedings from the IEEE SMC on Information Assurance Workshop (IAW)* (2005), pp. 42–49.

- [181] **Krause, J., Perer, A., and Stavropoulos, H.** Supporting Iterative Cohort Construction with Visual Temporal Queries. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 91–100.
- [182] **Kruskal, J., and Landwehr, J.** Icicle plots: Better Displays for Hierarchical Clustering. *The American Statistician* 37, 2 (1983), 162–168.
- [183] **Krzysztof, C.** Visualization as Support for Web Honeypot Data Analysis. *Information Systems in Management* 4, 1 (2015), 14–25.
- [184] **Kunz, T.** Visualizing Abstract Events. In *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research (CASCON)* (1994).
- [185] **Kwon, B. C., Verma, J., and Perer, A.** Peekquence: Visual Analytics for Event Sequence Data. In *Proceedings of the ACM Workshop on Interactive Data Exploration and Analytics (SIGKDD)* (2016), vol. 1.
- [186] **Lad, M., Massey, D., and Zhang, L.** Visualizing Internet Routing Changes. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (2006), 1450–1460.
- [187] **Lakkaraju, K., Bearavolu, R., Slagell, A., Yurcik, W., and North, S.** Closing-the-loop in NVisionIP: Integrating Discovery and Search in Security Visualizations. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 75–82.
- [188] **Lakkaraju, K., Yurcik, W., and Lee, A. J.** 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* (2004), pp. 65–72.
- [189] **Lam, H., Russell, D., Tang, D., and Munzner, T.** Session Viewer: Visual Exploratory Analysis of Web Session Logs. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)* (2007), pp. 147–154.
- [190] **Lamagna, W. M.** An Integrated Visualization on Network Events. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology Mini Challenge 2 (VAST)* (2011), pp. 319–321.
- [191] **Lancope.** Stealthwatch terminator. <http://www.lancope.com/products/>, 2015. Last Accessed: 30-05-2018.
- [192] **Landstorfer, J., Herrmann, I., Stange, J.-E., Dörk, M., and Wettach, R.** Weaving a Carpet from Log Entries: A Network Security Visualization built with Co-creation. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2014), pp. 73–82.
- [193] **Langner, R.** Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security & Privacy* 9, 3 (2011), 49–51.
- [194] **Lau, S.** The Spinning Cube of Potential Doom. *Communications of the ACM* 47, 6 (2004), 25–26.

- [195] **Lee, C. P., Trost, J., Gibbs, N., Beyah, R., and Copeland, J. A.** Visual Firewall: Real-time Network Security Monitor. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 129–136.
- [196] **Legg, P. A.** Visualizing the Insider Threat: Challenges and Tools for Identifying Malicious User Activity. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–7.
- [197] **Li, B., Springer, J., Bebis, G., and M.H. Gunes, M. H.** A Survey of Network Flow Applications. *Journal of Network and Computer Applications* 36, 2 (2013), 567–581.
- [198] **Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., and Tung, K.-Y.** Intrusion Detection System: A Comprehensive Review. *Journal of Network and Computer Applications* 36, 1 (2013), 16–24.
- [199] **Liao, Q., Shi, L., and Wang, C.** Visual Analysis of Large-scale Network Anomalies. *IBM Journal of Research and Development* 57, 3/4 (2013), 13–1.
- [200] **Liao, Q., Striegel, A., and Chawla, N.** Visualizing Graph Dynamics and Similarity for Enterprise Network Security and Management. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2010), pp. 34–45.
- [201] **Liu, X., Sun, Y., Fang, L., Liu, J., and Yu, L.** A Survey of Network Traffic Visualization in Detecting Network Security Threats. In *Proceedings of the International Conference on Trustworthy Computing and Services* (2014), pp. 91–98.
- [202] **Liu, Z., Wang, Y., Dontcheva, M., Hoffman, M., Walker, S., and Wilson, A.** Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 321–330.
- [203] **Livnat, Y., Agutter, J., Moon, S., Erbacher, R. F., and Foresti, S.** A Visualization Paradigm for Network Intrusion Detection. In *Proceedings from the IEEE SMC on Information Assurance Workshop (IAW)* (2005), pp. 92–99.
- [204] **Livnat, Y., Agutter, J., Moon, S., and Foresti, S.** Visual Correlation for Situational Awareness. In *IEEE Symposium on Information Visualization (InfoVis)* (2005), pp. 95–102.
- [205] **Makanju, A., Brooks, S., Zincir-Heywood, A. N., and Milios, E. E.** Logview: Visualizing Event Log Clusters. In *Proceedings of Conference on Privacy, Security and Trust (PST)* (2008), pp. 99–108.
- [206] **Malby, S., Mace, R., Holterhof, A., Brown, C., Kascherus, S., and Ignatuschtschenko, E.** Comprehensive Study on Cybercrime. *United Nations Office on Drugs and Crime, Tech. Rep* (2013).
- [207] **Maletic, J. I., and Marcus, A.** Data Cleansing: Beyond Integrity Analysis. In *IQ* (2000), pp. 200–209.

- [208] **Malik, S., Du, F., Monroe, M., Onukwugha, E., Plaisant, C., and Shneiderman, B.** Cohort Comparison of Event Sequences with Balanced Integration of Visual Analytics and Statistics. In *Proceedings of the International Conference on Intelligent User Interfaces (ICIUI)* (2015), pp. 38–49.
- [209] **Malik, S., Du, F., Monroe, M., Onukwugha, E., Plaisant, C., and Shneiderman, B.** Cohort Comparison of Event Sequences with Balanced Integration of Visual Analytics and Statistics. In *Proceedings of the International Conference on Intelligent User Interfaces* (2015), IUI '15, pp. 38–49.
- [210] **Mansmann, F., Keim, D. A., North, S. C., Rexroad, B., and Sheleheda, D.** Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1105–1112.
- [211] **Mansmann, F., Meier, L., and Keim, D. A.** Visualization of Host Behavior for Network Security. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSec)* (2008), pp. 187–202.
- [212] **Marchionini, G.** Exploratory Search: from Finding to Understanding. *Communications of the ACM* 49, 4 (2006), 41–46.
- [213] **Matuszak, W. J., DiPippo, L., and Sun, Y. L.** CyberSAVe: Situational Awareness Visualization for Cyber Security of Smart Grid Systems. In *Proceedings of the Workshop on Visualization for Cyber Security (VizSec)* (2013), pp. 25–32.
- [214] **Maynor, D.** *Metasploit toolkit for Penetration Testing, Exploit Development, and Vulnerability Research*. 2011.
- [215] **McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T., and Christensen, M.** Portvis: a Tool for Port-based Detection of Security Events. In *Proceedings of the ACM workshop on Visualization and Data Mining for Computer Security* (2004), pp. 73–81.
- [216] **Meyer, U., and Wetzel, S.** A Man-in-the-middle Attack on UMTS. In *Proceedings of the ACM workshop on Wireless Security* (2004), pp. 90–97.
- [217] **Microsoft.** Word Equation Editor. <http://support.office.com/en-US/article/Write-insert-or-change-an-equation-1D01CABC-CEB1-458D-BC70-7F9737722702/>, 2010.
- [218] **Microsoft.** Excel Advanced Criteria Filtering. <https://support.office.com/en-ie/article/Filter-by-using-advanced-criteria-4c9222fe-8529-4cd7-a898-3f16abdf32b/>, 2015. Last Accessed: 30-05-2018.
- [219] **Microsoft.** Wannacry and Petya. *Microsoft 0* (2017). Last Accessed: 30-05-2018.
- [220] **Mirkovic, J., and Reiher, P.** A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review* 34, 2 (2004), 39–53.

- [221] **Mitchell, R., and Chen, I.-R.** A Survey of Intrusion Detection Techniques for Cyber-physical Systems. *ACM Computing Surveys (CSUR)* 46, 4 (2014), 55.
- [222] **Mobley, R. K.** *An Introduction to Predictive Maintenance*. Butterworth-Heinemann, 2002.
- [223] **Monroe, M.** Interactive Event Sequence Query and Transformation. *PhD Thesis, University of Maryland* (2014).
- [224] **Monroe, M., Lan, R., Lee, H., Plaisant, C., and Shneiderman, B.** Temporal Event Sequence Simplification. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2227–2236.
- [225] **Monroe, M., Wongsuphasawat, K., Plaisant, C., Shneiderman, B., Millstein, J., and Gold, S.** Exploring Point and Interval Event Patterns: Display Methods and Interactive Visual Query. *Technical Report University of Maryland* (2012).
- [226] **Montigny-leboeuf, A. D., and Symchych, T.** Network Traffic Flow Analysis. In *Canadian Conference on Electrical and Computer Engineering* (2006), pp. 639–642.
- [227] **Mukhopadhyay, I., Gupta, K. S., Sen, D., and Gupta, P.** Heuristic Intrusion Detection and Prevention System. In *Proceedings of the International Conference and Workshop on Computing and Communication (IEMCON)* (2015), pp. 1–7.
- [228] **Mülders, D.** Network based Ransomware Detection on the Samba Protocol, 2017.
- [229] **Munzner, T.** A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009).
- [230] **Murray, D. G.** *Tableau your Data!: Fast and Easy Visual Analysis with Tableau Software*. John Wiley & Sons, 2013.
- [231] **Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S.** Malware Images: Visualization and Automatic Classification. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2011), pp. 4:1–4:7.
- [232] **Nativ, Y., and Shalev, S.** The Zoo. <https://github.com/ytisf/theZoo>, 2016–2017. Last Accessed: 30-05-2018.
- [233] **Nato.** The History of Cyber Attacks - A Timeline. <https://www.nato.int/docu/review/2013/cyber/timeline/en/index.htm>. Last Accessed: 24-05-2018.
- [234] **New York Times.** Hackers tried to cause Saudi petrochemical plant blast. <http://www.newindianexpress.com/world/2018/mar/16/hackers-tried-to-cause-saudi-petrochemical-plant-blast-new-york-times-1788227.html>, 2018. Last Accessed: 30-05-2018.
- [235] **Nielson, G., Hagen, H., and Muller, H.** Scientific Visualization. *Dagstuhl Seminar on Scientific Visualization* (1997).

- [236] **Nyarko, K., Capers, T., Scott, C., and Ladeji-Osias, K.** Network Intrusion Visualization with NIVA, an Intrusion detection Visual Analyzer with Haptic Integration. In *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS)* (2002), pp. 277–284.
- [237] **Nykodym, N., Taylor, R., and Vilela, J.** Criminal Profiling and Insider Cyber Crime. *Computer Law & Security Review* 21, 5 (2005), 408–414.
- [238] **Oberheide, J., Goff, M., and Karir, M.** Flamingo: Visualizing Internet Traffic. In *Proceedings of the Network Operations and Management Symposium (NOMS)* (2006), pp. 150–161.
- [239] **Onut, I.-V., and Ghorbani, A. A.** Svision: A Novel Visual Network-anomaly Identification Technique. *Computers & Security* 26, 3 (2007), 201–212.
- [240] **Orij, J.** Self-adaptation to Concept Drift in Web-based Anomaly Detection. *Master Thesis, University of Twente* (2016).
- [241] **Pang, R.** Towards Understanding Application Semantics of Network Traffic. *Thesis, Princeton University* (2008).
- [242] **Patcha, A., and Park, J.-M.** An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer Networks* 51, 12 (2007), 3448–3470.
- [243] **Pavlopoulos, G. A., Wegener, A.-L., and Schneider, R.** A Survey of Visualization Tools for Biological Network Analysis. *Biodata Mining* 1, 1 (2008), 12.
- [244] **Paxson, V., et al.** Guide, Bro Quick Start. <https://www.bro.org/sphinx-git/index.html>, 2015. Last Accessed: 30-05-2018.
- [245] **Pearlman, J., and Rheingans, P.** Visualizing Network Security Events using Compound Glyphs from a Service-oriented Perspective. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2008), pp. 131–146.
- [246] **Perer, A., and Wang, F.** Frequency: Interactive Mining and Visualization of Temporal Frequent Event Sequences. In *Proceedings of the International Conference on Intelligent User Interfaces* (2014), pp. 153–162.
- [247] **Perlin, K., and Fox, D.** Pad: an alternative Approach to the Computer Interface. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques* (1993), pp. 57–64.
- [248] **Phan, D., Paepcke, A., and Winograd, T.** Progressive Multiples for Communication-minded Visualization. In *Proceedings of Graphics Interface* (2007), pp. 225–232.
- [249] **Plackett, R.** Karl Pearson and the Chi-squared Test. *International Statistical Review* (1983), 59–72.

- [250] **Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B.** Lifelines: Visualizing Personal Histories. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1996), pp. 221–227.
- [251] **Pollitt, M.** An ad hoc Review of Digital Forensic Models. In *Proceedings of the International Workshop on Systematic Approaches to Digital Forensic Engineering* (2007), pp. 43–54.
- [252] **Popoola, S., Iyekekpolo, U., Ojewande, S., Sweetwilliams, F., and Atayero, A.** Ransomware: Current Trend, Challenges, and Research Directions. In *Proceedings of the World Congress on Engineering and Computer Science* (2017), vol. 1, pp. 169–174.
- [253] **Porter, T.** The Perils of Deep Packet Inspection. *Security Focus* (2005), 6.
- [254] **Postel, J.** User Datagram Protocol. <http://www.rfc-editor.org/rfc/rfc768.txt>, 1980. Last Accessed: 30-05-2018.
- [255] **Premathilaka, N. A., Aponso, A. C., and Krishnarajah, N.** Review on State of Art Intrusion Detection Systems designed for the Cloud Computing Paradigm. In *Proceedings of the International Carnahan Conference on Security Technology (ICCST)* (2013), pp. 1–6.
- [256] **Pretorius, A., and van Wijk, J.** What does the user want to see? What do the data want to be? *Information Visualization* 8, 3 (2009), 153–166.
- [257] **Pullen, M. J.** *Understanding Internet Protocols through Hands-on Programming*. John Wiley & Sons, Inc., 2000.
- [258] **Quinlan, J.** Induction of Decision Trees. *Machine learning* 1, 1 (1986), 81–106.
- [259] **Ramos, J., et al.** Using tf-idf to determine Word Relevance in Document Queries. In *Proceedings of the Instructional Conference on Machine Learning* (2003), vol. 242, pp. 133–142.
- [260] **Ren, P., Gao, Y., Li, Z., Chen, Y., and Watson, B.** IDGraphs: Intrusion Detection and Analysis using Histograms. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2005), pp. 39–46.
- [261] **Ren, P., Kristoff, J., and Gooch, B.** Visualizing DNS Traffic. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security* (2006), pp. 23–30.
- [262] **Reuille, T., Mahjoub, D., and Yan, P.** A Survey of Visualization Systems for Malware Analysis. In *Virus Bulletin OpenDNS* (2014), Virus Bulletin.
- [263] **Rind, A., Wang, T. D., Aigner, W., Miksch, S., Wongsuphasawat, K., Plaisant, C., Shneiderman, B., et al.** Interactive Information Visualization to Explore and Query Electronic Health Records. *Foundations and Trends® in Human-Computer Interaction* 5, 3 (2013), 207–298.

- [264] **Rogowitz, B., and Goodman, A.** Integrating Human-and Computer-based Approaches to Feature Extraction and Analysis. In *IS&T/SPIE Electronic Imaging* (2012), pp. 82910W–82910W.
- [265] **Romero-Gomez, R., Nadji, Y., and Antonakakis, M.** Towards designing effective Visualizations for DNS-based Network Threat Analysis. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2017), pp. 1–8.
- [266] **Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E.** SIP: Session Initiation Protocol. *Technical Report MIT, Columbia University* (2002).
- [267] **Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., and Schooler, E.** Session Initiation Protocol (SIP) RFC 3261. *IETF* (2002).
- [268] **Samba.** Samba Protocol. <https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>. Last Accessed: 30-05-2018.
- [269] **Scheepens, R., Michels, S., van de Wetering, H., and van Wijk, J. J.** Rationale Visualization for Safety and Security. In *Computer Graphics Forum* (2015), vol. 34, pp. 191–200.
- [270] **Schick, J., Wagner, M., Thür, N., Niederer, C., Rottermann, G., Tavalato, P., and Aigner, W.** Rule Ceation in a Knowledge-assisted Visual Analytics Prototype for Malware Analysis.
- [271] **Scott, D.** Scott’s rule. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 497–502.
- [272] **SecureList.** The Great Bank Robbery: The Carbanak APT. <https://securelist.com/the-great-bank-robbery-the-carbanak-apt/68732/>, 2015. Last Accessed: 30-05-2018.
- [273] **SHA.** Still Hacking Anyway. <https://sha2017.org/>, 2017. Last Accessed: 30-05-2017.
- [274] **Shiravi, H., Shiravi, A., and Ghorbani, A.** A Survey of Visualization Systems for Network Security. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1313–1329.
- [275] **Shiravi, H., Shiravi, A., and Ghorbani, A. A.** IDS Alert Visualization and Monitoring through Heuristic Host Selection. In *International Conference on Information and Communications Security (ICICS)* (2010), pp. 445–458.
- [276] **Shneiderman, B.** Dynamic Queries for Visual Information Seeking. *IEEE software* 11, 6 (1994), 70–77.
- [277] **Siadati, H., Saket, B., and Memon, N.** Detecting Malicious Logins in Enterprise Networks using Visualization. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), pp. 1–8.

- [278] **Siemens.** Industrial Communnication. <https://support.industry.siemens.com/cs/document/78028908?dti=0&lc=en-{U}{S}>, 2012. Last Accessed: 30-05-2018.
- [279] **Sikorski, M., and Honig, A.** *Practical Malware Analysis: the Hands-on Guide to Dissecting Malicious Software*. No Starch Press, 2012.
- [280] **Sloan, R.** Advanced Persistent Threat. *Engineering & Technology* 1, 1 (2014).
- [281] **Solarwinds.** Solarwinds Security Information and Event Management. www.solarwinds.com/Network/Software. Last Accessed: 24-05-2018.
- [282] **Solarwinds.** Voipmonitor. <http://www.voipmonitor.org/>, 2003. Last Accessed: 30-05-2018.
- [283] **Song, B., Choi, J., Choi, S.-S., and Song, J.** Visualization of Security Event Logs across Multiple Networks and its Application to a CSOC. *Cluster Computing* (2017), 1–12.
- [284] **Spence, R.** *Information Visualization*, vol. 1. 2001.
- [285] **Stack Exchange.** Why are terminal consoles still used. <https://ux.stackexchange.com/questions/101990/why-are-terminal-consoles-still-used>, 2018. Last Accessed: 30-05-2018.
- [286] **Stasiukonis, S.** Social engineering, the USB way. *Dark Reading* 7 (2006).
- [287] **Steinbach, M., Karypis, G., Kumar, V., et al.** A Comparison of Document Clustering Techniques. In *KDD workshop on text mining* (2000), vol. 400, pp. 525–526.
- [288] **Stokkel, M.** Ransomware Detection with Bro. https://www.bro.org/brocon2016/slides/stokkel_ransomware.pdf, 2016. Last Accessed: 30-05-2018.
- [289] **Stone, M.** In Color Perception, Size Matters. *IEEE Computer Graphics and Applications* 32, 2 (2012), 8–13.
- [290] **Syamkumar, M., Durairajan, R., and Barford, P.** Bigfoot: A Geo-based Visualization Methodology for Detecting BGP Threats. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), pp. 1–8.
- [291] **Tabish, S. M., Shafiq, M. Z., and Farooq, M.** Malware detection using Statistical Analysis of Byte-level file content. In *Proceedings of the ACM Workshop on CyberSecurity and Intelligence Informatics (SIGKDD)* (2009), pp. 23–31.
- [292] **Takada, T., and Koike, H.** Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs. In *Proceedings of the IEEE Symposium on Information Visualisation (InfoVis)* (2002), pp. 570–576.
- [293] **Tan, P.-N., et al.** *Introduction to Data Mining*. Pearson Education India, 2006.

- [294] **Taylor, T., Brooks, S., and McHugh, J.** Netbytes Viewer: An Entity-based Netflow Visualization utility for Identifying Intrusive Behavior. In *IEEE Workshop on Visualization for Computer Security (VizSec)* (2008), pp. 101–114.
- [295] **Taylor, T., Paterson, D., Glanfield, J., Gates, C., Brooks, S., and McHugh, J.** Flovis: Flow Visualization System. In *Proceedings of the Conference For Homeland Security, Cybersecurity Applications & Technology (CATCH)* (2009), pp. 186–198.
- [296] **TCPDump.** PCAP - Packet Capture Library. <https://www.tcpdump.org/manpages/pcap.3pcap.html>. Last Accessed: 30-05-2018.
- [297] **Tech Republic.** 5 Benefits of Command Line Tools. <https://www.techrepublic.com/blog/linux-and-open-source/five-benefits-of-command-line-tools/>, 2018. Last Accessed: 30-05-2018.
- [298] **Teoh, S. T., Ma, K.-L., Wu, S. F., and Jankun-Kelly, T.** Detecting Flaws and Intruders with Visual Data Analysis. *IEEE Computer Graphics and Applications* 24, 5 (2004), 27–35.
- [299] **Teoh, S. T., Ma, K. L., Wu, S. F., and Zhao, X.** Case study: Interactive Visualization for Internet Security. In *Proceedings of the Conference on Visualization* (2002), pp. 505–508.
- [300] **Teoh, S. T., Ranjan, S., Nucci, A., and Chuah, C.-N.** BGP eye: a new Visualization Tool for Real-time Detection and Analysis of BGP Anomalies. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security* (2006), pp. 81–90.
- [301] **Teoh, S. T., Zhang, K., Tseng, S.-M., Ma, K.-L., and Wu, S. F.** Combining Visual and Automated Data Mining for Near-Real-Time Anomaly Detection and Analysis in BGP. In *Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security* (2004), pp. 35–44.
- [302] **The Hacker News.** Smartinstall Hack in Cisco Routers. <https://thehackernews.com/2018/04/hacking-cisco-smart-install.html>, 2018. Last Accessed: 30-05-2018.
- [303] **The Wall Street Journal.** KPN admits to using Deep Packet Inspection. <https://blogs.wsj.com/tech-europe/2011/05/12/kpn-admits-to-using-deep-packet-inspection/>, 2011. Last Accessed: 30-05-2018.
- [304] **Theron, R., Magán-Carrión, R., Camacho, J., and Fernández, G. M.** Network-wide Intrusion Detection supported by Multivariate Analysis and Interactive Visualization. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2017), pp. 1–8.
- [305] **Tominski, C.** Event-Based Concepts for User-Driven Visualization. *Information Visualization* 10, 1 (2011), 65–81.

- [306] **Torvalds, L., and Hamano, J.** Git: Fast Version Control System. <http://git-scm.com>, 2010. Last Accessed: 30-05-2018.
- [307] **TrendMicro.** After Wannacry, UIWIX Ransomware and Monero-Mining Malware Follow Suit. <https://blog.trendmicro.com/trendlabs-security-intelligence/wannacry-uiwix-ransomware-monero-mining-malware-follow-suit/>, 2017. Last Accessed: 30-05-2018.
- [308] **Trostle, J., Van Besien, B., and Pujari, A.** Protecting against DNS Cache Poisoning Attacks. In *Proceedings of the IEEE Workshop on Secure Network Protocols (NPSec)* (2010), pp. 25–30.
- [309] **Tsymbal, A.** The problem of Concept Drift: Definitions and Related Work. *Computer Science Department Trinity College Dublin 106*, 2 (2004).
- [310] **Tulach, J.** *Practical API design: Confessions of a Java Framework Architect*. 2008.
- [311] **Turnbull, J.** *The Logstash Book*. 2013.
- [312] **Unger, A., Dräger, N., Sips, M., and Lehmann, D. J.** Understanding a Sequence of Sequences: Visual Exploration of Categorical States in Lake Sediment Cores. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 66–76.
- [313] **Van den Elzen, S., Holten, D., Blaas, J., and van Wijk, J. J.** Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 1–10.
- [314] **Van den Elzen, S., and Van Wijk, J. J.** Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2310–2319.
- [315] **Van der Aalst, W. M. P.** *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Publishing Company, Incorporated, 2011.
- [316] **Van der Aalst, W. M. P., de Leoni, M., and ter Hofstede, A. H.** Process Mining and Visual Analytics: Breathing Life into Business Process Models. *BPM Center Report BPM 17* (2011), 699–730.
- [317] **Van der Aalst, W. M. P., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., and Weijters, A. J. M. M.** Workflow Mining: A Survey of Issues and Approaches. *Data & knowledge engineering* 47, 2 (2003), 237–267.
- [318] **Van der Corput, P., and van Wijk, J. J.** ICLIC: Interactive Categorization of Large Image Collections. In *Proceedings of the IEEE Symposium on Pacific Visualization Symposium (PacificVis)* (2016), pp. 152–159.
- [319] **Van Dongen, B. F.** Business Processing Intelligence Challenge (BPIC). <http://www.win.tue.nl/bpi/doku.php?id=2011:challenge>, 2011. Last Accessed: 21-02-2017.

- [320] **Van Impe, K.** How STIX, TAXII and CybOX Can Help With Standardizing Threat Information. In *Professional Forum Security Intelligence* (2015).
- [321] **Virvilis, N., and Gritzalis, D.** The Big Four-What we did wrong in Advanced Persistent Threat Detection? In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)* (2013), pp. 248–254.
- [322] **VizSec.** IEEE International Symposium on Visualization for Cyber Security. <http://vizsec.org/>, 2017. Last Accessed: 30-10-2018.
- [323] **Vrotsou, K., Ellegard, K., and Cooper, M.** Everyday Life Discoveries: Mining and Visualizing Activity Patterns in Social Science Diary Data. In *Proceedings of the International Conference on Information Visualization (IV)* (2007), pp. 130–138.
- [324] **Vrotsou, K., Johansson, J., and Cooper, M.** Activitree: Interactive Visual Exploration of Sequences in Event-based Data using Graph Similarity. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 945–952.
- [325] **Wagner, D., and Soto, P.** Mimicry Attacks on Host-based Intrusion Detection Systems. In *Proceedings of the ACM Conference on Computer and Communications Security* (2002), pp. 255–264.
- [326] **Wagner, M., Fischer, F., Luh, R., Haberson, A., Rind, A., Keim, D., and Aigner, W.** A Survey of Visualization Systems for Malware Analysis. In *Eurographics Conference on Visualization (EuroVis) - STARS* (2015), R. Borgo, F. Ganovelli, and I. Viola, Eds., The Eurographics Association.
- [327] **Wagner, M., Rind, A., Thür, N., and Aigner, W.** A Knowledge-assisted Visual Malware Analysis System: Design, Validation, and Reflection of KAMAS. *Computers & Security* 67 (2017), 1–15.
- [328] **Walton, S., Maguire, E., and Chen, M.** Multiple Queries with Conditional attributes (QCATs) for Anomaly Detection and Visualization. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2014), pp. 17–24.
- [329] **Wang, T. D., Plaisant, C., Quinn, A. J., Stanchak, R., Murphy, S., and Shneiderman, B.** Aligning Temporal Data by Sentinel Events: Discovering Patterns in Electronic Health Records. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2008), pp. 457–466.
- [330] **Webga, K., and Lu, A.** Discovery of Rating Fraud with Real-time Streaming Visual Analytics. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2015), pp. 1–8.
- [331] **Wegman, E. J.** Hyperdimensional Data Analysis using Parallel Coordinates. *Journal of the American Statistical Association* 85, 411 (1990), 664–675.
- [332] **Wikipedia.** Linkedin Hack 2012. https://en.wikipedia.org/wiki/2012_LinkedIn_hack, 2018. Last Accessed: 30-05-2018.

- [333] **Willett, W., Heer, J., and Agrawala, M.** Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1129–1136.
- [334] **Wong, C.-K., Nutt, G., Jha, V., Sudarsanam, R., Papademetriou, S., and Aggarwal, A.** Delta Encoding using Canonical Reference Files, 2002. US Patent App. 10/117,006.
- [335] **Wong, T., Jacobson, V., and Alaettinoglu, C.** Internet Routing Anomaly Detection and Visualization. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)* (2005), pp. 172–181.
- [336] **Wongsuphasawat, K.** Interactive Exploration of Temporal Event Sequences. *PhD Thesis, University of Maryland* (2012).
- [337] **Wongsuphasawat, K., and Gotz, D.** Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2659–2668.
- [338] **Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., and Shneiderman, B.** Lifeflow: Visualizing an Overview of Event Sequences. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2011), pp. 1747–1756.
- [339] **Wongsuphasawat, K., Plaisant, C., Taieb-Maimon, M., and Shneiderman, B.** Querying Event Sequences by Exact Match or Similarity Search: Design and Empirical Evaluation. *Interacting with Computers* 24, 2 (2012), 55–68.
- [340] **Wongsuphasawat, K., and Shneiderman, B.** Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. In *Proceeding of the IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 27–34.
- [341] **Wüchner, T., Pretschner, A., and Ochoa, M.** Davast: Data-centric System Level Activity Visualization. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security (VizSec)* (2014), pp. 25–32.
- [342] **Xiao, L., Gerth, J., and Hanrahan, P.** Enhancing Visual Analysis of Network Traffic using a Knowledge Representation. In *Proceedings of the IEEE Symposium on Visual Analytics Science And Technology (VAST)* (2006), pp. 107–114.
- [343] **Yalcin, M. A., Elmqvist, N., and Bederson, B. B.** Keshif: Rapid and Expressive Tabular Data Exploration for Novices. *IEEE Transactions on Visualization and Computer Graphics* (2017).
- [344] **Yang, D., Usynin, A., and Hines, J. W.** Anomaly-based Intrusion Detection for SCADA Systems. In *International Topical meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (NPIC&HMIT)* (2006), pp. 12–16.
- [345] **Yelizarov, A., and Gamayunov, D.** Visualization of Complex Attacks and State of Attacked Network. In *Proceedings of the IEEE Workshop on Visualization for Cyber Security (VizSec)* (2009), pp. 1–9.

- [346] **Yeo, L. H., Che, X., and Lakkaraju, S.** Modern Intrusion Detection Systems. *CoRR abs/1708.07174* (2017).
- [347] **Yi, J. S., Kang, Y. A. H., and Stasko, J.** Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1224–1231.
- [348] **Yin, X., Yurcik, W., Treaster, M., Li, Y., and Lakkaraju, K.** VisFlowConnect: Cetflow Visualizations of Link Relationships for Security Situational Awareness. In *Proceedings of the ACM workshop on Visualization and Data Mining for Computer Security* (2004), pp. 26–34.
- [349] **Yüksel, O., den Hartog, J., and Etalle, S.** Reading Between the Fields: Practical, Effective Intrusion Detection for Industrial Control Systems. In *Proceedings of the ACM Symposium on Applied Computing* (2016), pp. 2063–2070.
- [350] **ZDNet.** Cyber-attacks are a top three risk to society. <https://www.zdnet.com/article/cyber-attacks-are-a-top-three-risk-to-society-alongside-natural-disaster-and-extreme-weather/>, 2018. Last Accessed: 30-05-2018.
- [351] **Zgraggen, E., Drucker, S. M., Fisher, D., and DeLine, R.** (S|qu)eries: Visual Regular Expressions for Querying and Exploring Event Sequences. *Technical Report Microsoft* (2015).
- [352] **Zhang, T., Liao, Q., and Shi, L.** Bridging the Gap of Network Management and Anomaly Detection through Interactive Visualization. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)* (2014), pp. 253–257.
- [353] **Zhang, T., Wang, X., Li, Z., Guo, F., Ma, Y., and Chen, W.** A Survey of Network Anomaly Visualization. *Science China Information Sciences* 60, 12 (2017), 121101.
- [354] **Zhang, Y., Xiao, Y., Chen, M., Zhang, J., and Deng, H.** A Survey of Security Visualization for Computer Network Logs. *Security and Communication Networks* 5, 4 (2012), 404–421.
- [355] **Zhao, J., Drucker, S. M., Fisher, D., and Brinkman, D.** Timeslice: Interactive Faceted Browsing of Timeline Data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (2012), pp. 433–436.
- [356] **Zhao, J., Liu, Z., Dontcheva, M., Hertzmann, A., and Wilson, A.** Matrixwave: Visual Comparison of Event Sequence Data. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2015), pp. 259–268.
- [357] **Zhao, Y., Liang, X., Fan, X., Wang, Y., Yang, M., and Zhou, F.** MVSec: Multi-perspective and Deductive Visual Analytics on Heterogeneous Network Security Data. *Journal of Visualization* 17, 3 (2014), 181–196.

- [358] **Zhong, C., Kirubakaran, D. S., Yen, J., Liu, P., Hutchinson, S., and Cam, H.** How to use experience in Cyber Analysis: An analytical reasoning support system. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI)* (2013), pp. 263–265.
- [359] **Zhou, F., Huang, W., Zhao, Y., Shi, Y., Liang, X., and Fan, X.** Entvis: A Visual Analytic Tool for Entropy-based Network Traffic Anomaly Detection. *IEEE Computer Graphics and Applications* 35, 6 (2015), 42–50.
- [360] **Zhu, X., and Goldberg, A.** Introduction to Semi-supervised Learning. *Synthesis lectures on artificial intelligence and machine learning* 3, 1 (2009), 1–130.

Summary

Interactive Visualization of Event Logs for Cybersecurity

In our ever-growing world of interconnected devices and processes, the amount of data transferred in these networks is exploding. With cybercrime as the number one threat in our ICT society we need to understand how and for what purpose networks are used in order to protect them from misuse. The discovery of undesired behavior in environments can help to detect and prevent cybercrime organizations from destroying and abusing our infrastructures.

Many domains analyze network environments by logging their behavior in events. Examples are messages sent between mobile devices, medical treatments taken in a hospital, or financial transactions paid with your credit card. The discovery of malicious patterns and anomalies in real-world event data however is nontrivial as they often contain a wide variety of metadata depending on the domain. Fully automated anomaly detection techniques suffer from many errors in this high-dimensional data due to the lack of context whereas manual analysis of this data is often unfeasible due to its size and variety.

In this dissertation we explore and present interactive visualization techniques to explain and detect outliers (also known as anomalies) in large event collections. In particular, we focus on the following research question:

How can we use interactive visualization techniques and automated methods to discover relevant patterns and anomalies in large event collections?

We show how we can combine visualization and automated techniques to support the exploration and explanation of different classes of patterns and anomalies. For each of these classes we introduce novel interaction and query mechanisms where we combine visualization with automatic techniques from various domains including data-mining, process-mining, and language engineering.

In Chapter 3 we present a novel exploration method on how to discover anomalous events by converting the event metadata to a pixel visualization. Combined with an online classifier, parts of the metadata are lit up whenever events contain values that are classified as malicious. Through interaction users are enabled to explore the validity of the metadata attribute space and refine classification results where necessary.

When trying to assess the relevance of an anomaly, context plays an important role. For instance, although the access of a file X does not have to be malicious in general, it can be considered dangerous when performed by a certain user. The way we split our data therefore determines the type of anomalies that we can discover. In Chapter 4 we present a system to

inspect alert data from different perspectives. We show how visualization and interaction can be used to enable analysts to discover high-level threats in a collection of low-level alert collections.

Chapter 5 focuses on the analysis of anomalies in event sequences. In this chapter we present a system called Eventpad that enables rapid and cost-effective discovery of patterns in event collections by visualizing them as blocks on a screen. Rules enable users to highlight and visual encode event properties that are of interest. Automated techniques such as clustering and alignment in turn can use this labeling to discover patterns between event sequences. Similar to a notepad editor, find & replace functionality and conditional formatting can be used to quickly search and highlight outliers in the data.

In Chapters 6 and 7 we show how we can apply our techniques to discover illegal traffic activity in a wildlife preserve and the analysis of ransomware activity. The use cases in Chapters 5 and 6 show that the problems and techniques described in the dissertation do not limit themselves to the analysis of network traffic for cybersecurity, but in general are applicable to tabular data from any domain.

Samenvatting

Interactieve Visualisatie van Systeemgebeurtenissen voor Cybersecurity

In onze samenleving worden steeds meer apparaten en processen met elkaar verbonden. De hoeveelheid informatie dat in deze netwerken wordt verstuurd is aan het exploderen. Met cybercrime als de nummer 1 bedreiging in onze ICT samenleving moeten we begrijpen hoe en waarvoor onze netwerken worden gebruikt om enige vorm van misbruik te voorkomen. De ontdekking van kwaadaardig gedrag kan bijvoorbeeld helpen bij het opsporen van cybercrime organisaties en het voorkomen van misbruik of zelfs vernietiging van eigendommen.

Veel bedrijven en domeinen analyseren hun netwerkomgevingen door systeemgedrag in gebeurtenissen op te slaan. Denk hierbij bijvoorbeeld aan het versturen van berichten tussen mobiele telefoons, het opslaan van medische behandelingen in elektronische patientendossiers en het bijhouden van financiële creditcard transacties. Het ontdekken van ongewenste patronen of gevaarlijke afwijkingen in dit soort data is in praktijk vaak moeilijk door de grote variatie in deze gegevens. Daarnaast kunnen deze gegevens zwaar verschillen van domein tot domein. Volledig automatische oplossingen maken vaak te veel fouten om kwaadaardig gedrag in deze hoog-dimensionale data te vinden door gebrek aan context terwijl handmatige analyse van deze gegevens vaak onmogelijk is door de grootte en variatie.

In dit proefschrift introduceren en demonstreren we interactieve visualisatietechnieken om in grote collecties gebeurtenissen patronen en afwijkingen (ookwel anomalieën genoemd) te ontdekken. Hierin staat de volgende onderzoeksvraag centraal:

Hoe kunnen we interactieve visualisatie technieken en automatische methoden gebruiken om relevante patronen en anomalieën te vinden in grote collecties gebeurtenissen?

In dit proefschrift laten we zien hoe we visualisatie en automatische technieken kunnen gebruiken om verschillende typen patronen en afwijkingen te kunnen zoeken en verklaren. Voor iedere type afwijking introduceren we nieuwe interactie- en zoekmechanismen door visualisatie te combineren met automatische technieken uit verschillende domeinen zoals data-mining, process-mining en language engineering.

In Hoofdstuk 3 presenteren we een nieuwe werkwijze om afwijkende gebeurtenissen te vinden door zogenaamde metadata uit gebeurtenissen te extraheren en deze om te zetten in een pixel visualisatie. Met behulp van een online classificatie algoritme worden delen van de metadata automatisch opgelicht bij de ontdekking van kwaadaardige patronen. Door middel van interactie kunnen gebruikers de relevantie van deze alarmen analyseren en waar nodig het classificatie algoritme afstellen.

Om de ernst van een afwijking te bepalen, speelt context een belangrijke rol. Hoewel de toegang tot een bestand X in het algemeen niet kwaadaardig hoeft te zijn, kan deze wel als gevaarlijk worden beschouwd als dit door een bepaalde gebruiker wordt uitgevoerd. Het perspectief waarmee we naar de data kijken is dus bepalend voor het soort afwijkingen die we kunnen ontdekken. In Hoofdstuk 4 laten we zien hoe we visualisatie en interactie kunnen gebruiken om grote collecties automatisch geclassificeerde afwijkingen snel te verkennen door de data vanuit verschillende invalshoeken te inspecteren. Hiermee kunnen we laten zien hoe we bedreigingen op netwerk niveau kunnen herleiden uit een collectie van abstracte alarmen op gebeurtenis niveau.

Hoofdstuk 5 richt zich op de analyse van afwijkingen in reeksen van gebeurtenissen. In dit hoofdstuk presenteren we een systeem genaamd Eventpad om snel en effectief patronen in gebeurtenissen te vinden door deze te visualiseren als blokjes op een scherm. Met behulp van regels kunnen gebruikers bepaalde eigenschappen in gebeurtenissen visueel benadrukken om daarmee ongewenste patronen te ontdekken die relevant zijn voor hun onderzoek. Automatische methoden zoals clustering en alignment kunnen worden gebruikt om op basis van deze visuele representatie patronen tussen gebeurtenisreeksen te ontdekken. Net als in een tekstverwerker kunnen we find & replace functionaliteit en conditionele opmaak gebruiken om snel afwijkingen te zoeken en in de data zichtbaar te maken.

In Hoofdstukken 6 en 7 laten we zien hoe we de nieuwe technieken kunnen gebruiken om illegale verkeerspatronen in een natuurreserveaat te ontdekken en snel ransomware activiteit kunnen analyseren. De toepassingen van de systemen in Hoofdstukken 5 en 6 laat zien dat de voorgestelde technieken zich niet beperken tot de analyse van netwerkverkeer, maar deze ook kunnen worden gebruikt voor de analyse van tabelgegevens uit andere domeinen.

Curriculum Vitæ

Bram Cornelis Maria Cappers was born on June 10 1991 in Heerlen, the Netherlands. After his secondary education in 2009 at the Grotius College in Heerlen, he started the Computer Science program at the department of Mathematics and Computer Science at Eindhoven University of Technology. During this time he was already active as a developer and consultant in the area of web engineering and software development for enterprise automation. In 2012 he obtained his Bachelor degree with honors and started the Computer Science and Engineering program at the same university. In 2014 Bram obtained his Master of Science degree with honors and completed the TU/e Honors program. During his graduation in the Software Engineering group, Bram worked on the visualization and explanation of the generalized LL parsing algorithm under guidance of prof. dr. Mark van den Brand and prof. dr. ir. Jarke J. van Wijk.

In September 2014 he started his PhD under supervision of prof. dr. ir. Jarke J. van Wijk and prof. dr. S. Etalle in the area of Data Visualization for Cybersecurity. In the project SpySpot he developed new techniques to visually detect (or aid in the detection of) cyber espionage and targeted malware inside computer networks. The results of the SpySpot project, completed in 2018, were published in a number of scientific international conference proceedings and journals. In addition, the technology has been published at various high-rated industrial reports and events including Still Hacking Anyway 2017 [273], European Cyber Security Perspectives 2018 [179], and Black Hat USA 2018 [29].

During his PhD, Bram was an active member in the Program Committee of the International Symposium on Visualization for Cyber Security (VizSec). In 2015 he started an internship at Motto Communications to study fraud in Voice Over IP telephony traffic. Outside academia, Bram worked as a volunteer in the development and organization of a (semi-)yearly gaming event TiMeS for up to 250 people. In 2018 he became a member of KPN's Computer Emergency Response Team for the analysis of malicious network activity. In addition, Bram received three awards and an NWO Demonstrator Grant for his system Eventpad. At VAST 2017 he won the IEEE award "Elegant tool for hypothesis testing and generation". In 2018 Eventpad also won the ICT.Open Best Demo award for "promising Dutch ICT technology of 2018". As a future entrepreneur, Bram won the best Executive Summary award at the European Venture Program 2018.

10-06-1991 Born in Heerlen, The Netherlands.

Education

- 2003–2009 Grammar School
Grotius College Atheneum, Heerlen
- 2009–2012 Undergraduate in Computer Science Bachelor
Eindhoven University of Technology
Department Mathematics & Computer Science
Graduation with honors (8.0 out of 10.0)
- 2012–2014 Undergraduate in Computer Science & Engineering Master
Eindhoven University of Technology
Department Mathematics & Computer Science
Thesis: Exploring and Visualizing GLL parsing
Supervisors: prof. dr. Mark v.d. Brand,
prof. dr. ir. Jarke J. van Wijk
Graduation with honors (8.9 out of 10.0)
Completion of the TU/e Honors Program
- 2014–2018 PhD Candidate Computer Science
Eindhoven University of Technology
Department Mathematics & Computer Science
Data Visualization group
Thesis: Interactive Visualization of Event Logs for Cybersecurity
1st promotor: prof. dr. ir. Jarke J. van Wijk
2nd promotor: prof. dr. Sandro Etalle

Awards

- 2017 IEEE Visual Analytics Science and Technology (VAST) Challenge 2017 award
Elegant tool for Hypothesis Testing and Generation
- 2018 ICT.Open 2018 award *Best Demo*
- 2018 European Venture Program 2018 Scholarship award *Best Executive Summary*
- 2018 NWO Demonstrator Grant 2018 for
“Scaling-up Eventpad: Rapid and Cost-effective Cyber Security Monitoring”

In our ever growing world of interconnected devices and processes, the amount of data transferred in these networks is exploding. With cybercrime as the number one threat in our ICT society we need to understand how and for what purpose networks are used in order to protect them from misuse. The discovery of undesired behavior in environments can help to detect and prevent cybercrime organizations from destroying and abusing our infrastructures.

In this dissertation we explore and present interactive visualization techniques to explore and detect outliers (also known as anomalies) in large event collections. In particular, we focus on the following research question:

How can we use interactive visualization techniques and automated methods to discover relevant patterns and anomalies in large event collections?

We show how we can combine visualization and automated techniques to support the exploration and explanation of different classes of patterns and anomalies. For each of these classes we introduce novel interaction and query mechanisms where we combine visualization with automatic techniques from various domains including data-mining, process-mining, and language engineering.



ISBN 978-94-6380-043-3

Finding anomalies is not difficult
The challenge is finding the ones that matter

