Faculty of Information Technology
Brno University of Technology

Visual Analytics for Cybersecurity Education and Training

Habilitation Thesis
(Collection of Articles)

Radek Ošlejšek

January 2022
Brno, Czech Republic

Abstract

The increasing number of security threats leads to a growing need to develop new methods for their mitigation. Simultaneously, it is necessary to train more and more experts who would recognize these threats in time. However, comprehension and practical training of cyber-defense processes is challenging. It is not possible to utilize production infrastructures as these activities would endanger them. Instead, it is necessary to use isolated environments emulating real critical infrastructures. Progress in the development of suitable environments occurred only recently and is associated with the expansion and maturity of cloud technologies. However, the availability of suitable cybersecurity platforms is only one piece of the puzzle. A deeper understanding of cybersecurity processes requires employing efficient data analysis methods capable of providing insight into relationships hidden in the data. In our research, we deal with both interconnected areas. We aim to develop a suitable environment, where security experiments and practical training can be conducted, and relevant data can be systematically gathered. Simultaneously, we strive to use the data for threats understanding and training mitigation procedures. We use the exploratory, visual-based approaches to the data analysis.

In this thesis, I aim to provide the readers with a comprehensive overview of our results in the field of cybersecurity training platforms and related analytical visualizations that we reached in the last seven years. The thesis is structured as a collection of relevant papers accompanied by a commentary putting our contributions in the context of the state-of-the-art in the area and summarizing our achievements. The thesis consists of two main parts. In the first part, I focus on the different approaches to education and training. Our cloud-based cyber range is presented. Lessons learned from the utilization of the platform for various types of cyber exercises are discussed. The second part contains our achievements in the field of visualizations and exploratory data analysis. Conceptual works mapping the possibilities of visual-analysis methods in this new application domain are presented. Particular visualizations improving the efficiency of hands-on training programs are discussed as well. Our achievements in the forensic investigation of file system metadata are presented along with learning analytics results.

Keywords: Cyber security, cyber range, exercise, training, analysis, visual analytics, visualization.

Abstrakt

S tím, jak společnost čelí narůstajícímu počtu bezpečnostních hrozeb, narůstá i potřeba vyvíjet nové postupy k jejich potlačování. Zároveň je nutné trénovat stále více expertů, kteří by byli schopni tyto hrozby včas rozpoznat a účinně se jim bránit. Pochopení bezpečnostních postupů a jejich praktické natrénování přitom představují obrovský problém. Aby nebyla ohrožena provozní infrastruktura, není možné výcvik provádět v reálných počítačových sítích. Je nutné využívat izolovaná prostředí emulující kritické infrastruktury. Vývoj takových prostředí je velmi náročný. Výrazný rozvoj potřebných platforem nastal až posledním období a je spojen s rozvojem cloudových technologií. Samotná platforma ovšem nestačí. K jejímu maximálnímu využití jen nutné umět analyzovat bezpečnostní data a poskytovat vhled do jejich skrytých vazeb tak, aby bezpečnostní experti byli schopni pochopit nové hrozby, rozpoznat je, a nacvičovat obranné postupy. Ve svém výzkumu se proto zabýváme oběma propojenými oblastmi. Snažíme se vyvíjet vhodné prostředí pro bezpečnostní experimenty a praktický výcvik a zároveň usilujeme o využití získaných dat k porozumění bezpečnostních hrozeb a trénování postupů. Zaměřujeme se přitom hlavně na využití vizuálně-analytických přístupů.

Tato práce si klade za cíl seznámit čtenáře s uceleným přehledem našich výsledků v oblasti vývoje bezpečnostních platforem a souvisejících analytických vizualizací, kterých jsme za posledních sedm let našeho působení v této oblasti dosáhli. Práce je souborem mých relevantních vědeckých publikací doprovozených komentářem, který mé výsledky zasazuje do kontextu aktuálního stavu výzkumu v dané oblasti. Práce je rozdělena na dvě hlavní části. V první části se zaměřuji na různé přístupy k výuce a tréninku počítačové bezpečnosti. Představuji naše vlastní řešení v podobě moderní cloudové platformy a shrnuji zkušenosti s jejím použitím pro různé typy praktického výcviku. Druhá část práce obsahuje naše výsledky z oblasti vizualizací a explorativní analýzy. Jsou představeny koncepční práce mapující možnosti vizuálně analytických metod v této nové aplikační doméně. Rovněž jsou popsány naše konkrétní vizualizace, které jsme dosud vyvinuli pro potřeby zefektivnění praktického výcviku. Kromě výsledků z oblasti výuky je představen také interaktivní nástroj pro praktickou forenzní analýzu metadat disků.

Klíčová slova: Počítačová bezpečnost, bezpečnostní platforma, výcvik, analýza, vizuální analýza, vizualizace.

Acknowledgements

I would like to express my appreciation to all the mentors I have had along my journey—to Jiří Sochor for guiding me during my Ph.D. studies, to Ivan Kopeček for helping me to dive into the field of data semenantics modeling, and to Tomáš Pitner for welcoming me warmly in the LaSArIS lab after I joined FI MU as an assistant professor. I would like to thank all my colleagues, co-authors, LaSArIS members, and KYPO project participants for being such a great team to work with.

Finally and foremost, I wish to thank my family and close friends for their support, understanding, and patience.

Radek Ošlejšek

Contents

Part I

Commentary

Chapter 1

Introduction

A shortage of cybersecurity workforce poses a critical danger for current companies and nations [191, 164]. As modern society is exposed to the increasing number of cyber threats, there is a growing need to train new cybersecurity experts.

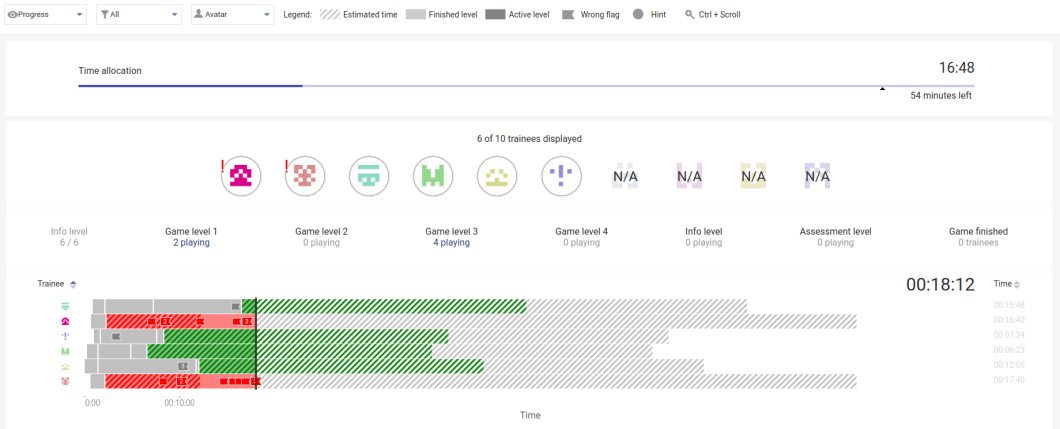

Figure 1.1: An example of visual analytics for hands-on training: Situational awareness for tutors.

Operational environments are not suitable for building a systematic knowledge of new cyber threats and training responses. In the last decade, a big effort has been made to design and implement *cyber ranges* that would provide places to practice skills such as network defense, attack detection and mitigation, penetration testing, and many others in a realistic environment. They serve as isolated, realistic environments in which security and operations teams can be trained without the risk of endangering real computer networks.

Along with the development of cyber ranges, there has been a significant increase in

hands-on competitions, challenges, and exercises. The two most widely recognized hands-on cybersecurity training types are *Capture the Flag* (CTF) games [251, 236, 59, 239] and *Cyber Defense eXercises* (CDX) [185]. The main difference lies in their educational goals and complexity. CTF games focus primarily on the cybersecurity skills of future experts. They are well structured and then often used for regular courses. On the contrary, CDXs have been traditionally organized by military and governmental agencies. They emphasize realistic training scenarios that authentically mimic the operational environment of a real organization. For these reasons, every new CDX is unique, and its preparation in a cyber range still requires a considerable amount of skills and workforce.

Regardless of the training type, learning analysis of cybersecurity evens is always difficult. In many IT areas, hands-on training produces a tangible output. For example, the output of a programming course is a code that can be checked and assessed. On the contrary, cybersecurity is process-oriented. The learner's goal is to scan the computer network, find a vulnerable server, and exploit the vulnerability to hack the server, for instance. Modern cyber ranges are able to collect data related to the behavior of learners and the state of network infrastructure. However, their analysis represents a challenging task. It is difficult to reconstruct processes and provide meaningful analytical views of learning aspects like "what is the most difficult part of the exercise" or "what is the common mistake of learners in solving tasks". In our research, we put emphasis on the support of learning and behavior analysis aiming to provide analytical tools that would help learners in gaining insight into the complex cybersecurity processes and tutors in improving the impact of exercises.

Visualizations represent a widely adopted method of data exploration and analysis. In 1996, Ben Shneiderman [214] addressed the problem of information overload in data visualization, formulating the "information-seeking mantra": overview first, zoom and filter, then details-on-demand. Approaches to visual analytics [253] cover the complete analytical reasoning process supported by interactive visual interfaces [199]. They are applied in various fields, from biology or weather forecasts [132, 129, 64, 65] to education [232, 230]. Our research activities focus on the application of visual-analytics methods on data analysis in cybersecurity training and forensic investigation.

1.1 Focus and Outline of the Thesis

This thesis summarizes my contributions to the design of cyber ranges and their analytical features. Along with cybersecurity experts, experts on the simulation of computer networks in clouds, and visual analysts, we reached within the last decade several achievements contributing to the organization of practical training programs and supporting exploratory learning analysis in this domain.

In the text, I first focus on the architecture of modern cyber ranges and lessons learned from the organization of cybersecurity training programs of various types. This thesis builds

on the *KYPO Cyber Range* [243], which has been in development at the Masaryk University since 2013. The results are described in Chapter 2. In Chapter 3, I present our achievements in visual analytics, where we contributed to both learning analysis of hands-on training programs and forensic investigation.

Each of the two chapters introduces the reader to the state-of-the-art in the respective field, summarizes our contribution, and maps information into relevant articles that I have co-authored. The overall collection of the articles is listed in Part II of this thesis to exemplify my contributions.

Chapter 2

Hands-on Cybersecurity Training

2.1 Training Platforms

Operational networks are not suitable for training responses to cyber threats because any mistake, intentional or unintentional, can damage the infrastructure. Therefore, cyber ranges or testbeds are used for this purpose. These are usually built to provide secure virtual environments where the cybersecurity process can be monitored, studied, and analyzed without the risk of threatening operational infrastructure or where users can learn how to defend their systems against threats and attacks.

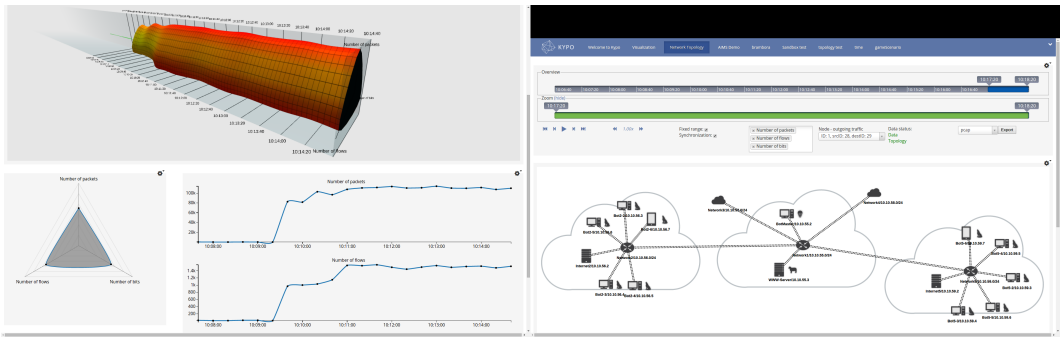

Figure 2.1: Web user interface of the KYPO Cyber Range.

Cybersecurity platforms can be divided into three basic categories, each reflecting specific purposes of the cybersecurity domain: generic testbeds, lightweight platforms for cybersecurity training, and cyber ranges.

Generic testbeds provide basic functionality for the emulation of computer networks. *Emulab/Netbed* [252] has been developed since 2000 and can be considered as a prototype of an emulation testbed for research into networking and distributed systems. It allocates computing resources for a specified network and instantiates the network at a dedicated hardware infrastructure. It provides accurate, repeatable results in experiments with moderate network loads [216, 183]. Another representative of this category, *CyberVAN* [3], is an experimentation testbed with hybrid emulation providing the ability to dynamically reconfigure the simulated network and the host nodes. It is able to simulate large strategic networks approximating large ISP networks and employs Big Data Analytics engines and techniques for post-mortem analysis.

Lightweight platforms have been developed primarily for cybersecurity training. While some of them evolved from generic testbeds, others were designed from scratch with different needs in mind. *Avatao* [39, 1] is a web-based online e-learning platform offering IT security challenges (hands-on exercises), which can be organized to a path which leads to fulfilling an ultimate learning objective. In the *Hacking-Lab* [210] online platform, teams of participants have to perform several tasks simultaneously. Many lightweight platforms [56, 236, 189] focus on capture-the-flag games, which are similar to multi-level computer games where participants perform cyber-security tasks prescribed by individual levels, e. g. scan the network, find a vulnerable server, overtake the server.

Cyber ranges are complex virtual environments that are used not only for cyberwarfare training but also for cyber technology development, forensic analysis, and other cyber-related issues. One very popular cyber range is *DETER/DeterLab* [165, 23], which is based on Emulab and was started with the goal of advancing cybersecurity research and education in 2004. There are currently many other cyber ranges, e. g., *National Cyber Range (NCR)* [83, 172], *Michigan Cyber Range (MCR)* [156], *SimSpace Cyber Range* [194], *EDURange* [4], *CyRIS* [186], *CyTrONE* [27], or *CYRAN* [101].

A comprehensive survey of state-of-the-art cyber ranges and testbeds [60] published by the Australian Department of Defence in 2013 shows that our research started at the time when the concept of generic cyber ranges was in its initial stages. That was the reason we decided to develop our research platform, which we named *KYPO Cyber Range*. Our cyber range is based on several principles:

- *Flexibility of Network Management* – computer networks are fully virtualized in a cloud. For the topology nodes, a wide range of operating systems is supported (including arbitrary software packages). Network connections are emulated. Cloud-based virtualization brings the possibility to instantiate networks on-demand, clone them, align their parameters, and other dynamic aspects of network management.
- *Isolation* – network topologies and platform users can be isolated from the outside world and each other so that experiments and cyber-related activities cannot threaten other users or infrastructures.

- *Interoperability* – in contrast to isolation, integration with (or connection to) external systems is also achievable with reasonable effort. For example, it is possible to connect an existing physical computer to the virtualized computer network.
- *Build-In Monitoring and Data Gathering* – the platform natively provides both real-time and post-mortem access to detailed monitoring data. These data are related to individual topologies, including flow data and captured packets from the network links, as well as node metrics and logs. The monitoring subsystem is flexible, enabling us to gather heterogeneous data and adapt the monitoring to specific requirements of cyber scenarios or analytical tasks.
- *Easy Access* – users with a wide range of experience should be able to use the platform. For less experienced users, web-based access to its core functions is available. Expert users, on the other hand, can interact with the platform via advanced means, e.g., using remote SSH access.
- *Providing Insight and Analytical Tools* – the primary goal of cyber ranges is to support users in gaining insight into complex cybersecurity processes. Therefore, the KYPO Cyber Range puts great emphasis on providing exploratory visualizations and user interfaces that would be able to mediate the semantics of cybersecurity data to users, support situational awareness of developments in the cyber range, and support analytical tasks.

These features are often contradictory, which makes the design and architecture of the platform ambitious and challenging. On the other hand, they enable us to use the platform for a wide variety of cybersecurity tasks, including training, forensic analysis, and cybersecurity research.

Article C: Vykopal, J., Ošlejšek, R., Čeleda, P., Vizváry M., Tovarňák, D.: KYPO Cyber Range: Design and Use Cases. In *International Conference on Software Technologies (ICSOFT'17)*. Madrid, Spain: SciTePress, 2017. p. 310–321, 12 pp.

Contribution (20%): I coordinated the design of the system architecture. I contributed to the data monitoring and management components and was responsible for the design and development of user interfaces and interactions. I wrote corresponding parts of the paper.

Publication type: Conference, CORE rank B.

The architecture and design decisions made during the development are summarized in **Article C**. Non-trivial engineering work resulted in a component-based, highly distributed platform. We operate an instance of the cyber range at Masaryk University since 2014. A new generation of the training platform was published as open-source in 2021. It is the first publicly available cyber range worldwide.

Advanced networking is one of the most important features of KYPO. Computer networks and other critical infrastructures are emulated on demand in a cloud. The underlying cloud infrastructure uses IEEE 802.1Q, i.e., Virtual LAN tagging, using Q-in-Q tunneling. Therefore, multiple cloud providers are supported. Moreover, KYPO also enables to connect systems and devices that do not have a virtualized operating system, i.e. they are hardware-dependent or location dependent. It is possible to connect an existing PC or subnet into the virtualized infrastructure. This feature is useful for forensic investigation and cybersecurity research. For example, a suspicious device can be connected to the isolated virtualized network with preinstalled analytical tools, and the communication of the device over the network can be safely examined.

The cyber range is equipped with comprehensive monitoring and data gathering subsystem. The monitoring management component provides fine-grained control over the built-in monitoring configuration and provides an API that exposes the acquired monitoring data to external consumers. Heterogeneous data like user interactions, history of shell commands, or the state of network nodes and links can be collected and utilized either at run-time for situation awareness or after experiments and training sessions.

A web portal mediates access to the platform for the end-users by providing them with interactive user interfaces. In particular, the web portal is designed to deal with the management of cyber exercises, role-based access control to virtualized computer networks, data analysis, and situational awareness.

Article D: Eichler, Z, Ošlejšek, R., Toth, D.: KYPO: A Tool for Collaborative Study of Cyberattacks in Safe Cloud Environment. In *HCI International 2015: Human Aspects of Information Security, Privacy, and Trust.* LNCS, vol. 9190. Los Angeles: Springer International Publishing, 2015. p. 190–199, 10 pp.

Contribution (35%): I was responsible for the coordination of design and implementation activities of the interactive visual platform. I supervised a team of developers and participated in writing the paper.

Publication type: Conference|Book chapter, Springer.

In spite of the general purpose of KYPO, the cyber range is primarily used as a training platform. To support the organization of both CTF games and CDXs, we had to analyze corresponding organizational processes, clarify learning objectives, and formalize data. This research led to the design and development of user interfaces that automatize often repeated tasks and provide insight into training sessions.

In **Article D**, we describe the basic principles that we used for the design of a highly collaborative environment. Different collaboration modes are discussed, reflecting the need for sharing and duplicating data for various collaboration scenarios. This conceptualization

affected the overall architecture of KYPO, mainly the data storage subsystem and the design of user interfaces.

In order to cope with the largely heterogeneous monitoring data, we use the normalized design pattern and the notion of a monitoring bus component implementing this pattern, as described in detail by [227]. The long-term objective of such a deployment is to render the monitoring architecture within the platform fully event-driven. This is motivated by the growing need for advanced monitoring data correlations both in terms of real-time and post-mortem analysis.

2.2 Training Content

Cybersecurity skills require higher-order thinking. The best way to develop and ameliorate these abilities is through practical hands-on courses [157, 155]. It is believed that they enable participants to effectively gain or practice diverse security skills in a fun way.

One of the commonly used learning methods for training problem-solving or various IT skills (e.g., programming) is puzzle-based learning. Michalewicz et al. [161] introduced a game-based learning method that uses puzzles as a metaphor for getting students to think about how to frame and solve unstructured problems. In IT education, the puzzle-based learning approach is prevalent for many years [258, 159, 104]. Multiple studies confirmed the usefulness of this approach also for cybersecurity [92, 73, 107, 58] education.

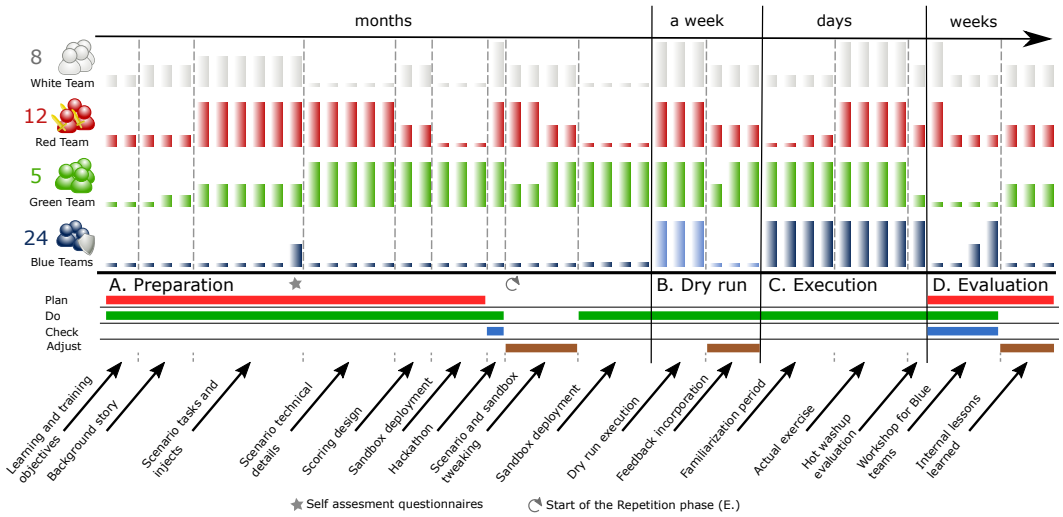

Figure 2.2: The life cycle of a cyber defense exercise. KYPO supports even such complex training events.

Puzzle-based learning in the cybersecurity domain is primarily represented by CTF games

(see Chapter 1). CTF training scenarios serve as puzzle-based templates structuring the content into levels focused on solving cybersecurity tasks, e.g., scan the network, identify a server, find the server vulnerability, exploit it, and gain the root privileges. Finding a level solution is necessary to proceed to the next one. Trainees are penalized when taking hints or solutions and reach score points for successful solutions. CTF games are well-structured and then well supported by training platforms, including KYPO. KYPO aims to automatize the entire CTF life cycle, making the organization of training programs routine. It is possible to prepare CTF content via web user interfaces, invite and manage participants, allocate cloud resources, organize and supervise training sessions, and also analyze collected data, as discussed in Chapter 3. Nowadays, the Faculty of Informatics, Masaryk University, organizes regular cybersecurity courses of its curriculum in this form [239, 245].

On the contrary, CDXs aim to simulate real conditions. They are more complex and often designed from scratch. Therefore, the conceptualization of CDXs and their unification is difficult. Only a few public research papers have dealt with the design of an exercise in a cyber range. Granåsen and Andersson conducted a case study on measuring team effectiveness in Baltic Cyber Shield 2010, a multi-national civil-military CDX [95]. The Spanish National Cybersecurity Institute proposed a taxonomy of cyber exercises [71] that recognizes operations-based exercises focused on the incident response by participants in technical and management roles. The ISO/TC 223 effort resulted in ISO 22398, which describes general guidelines for exercises, including basic terms and definitions [117]. Unfortunately, the implementation details of an exercise in a cyber range are beyond the scope of this standard.

Article B: Vykopal, J, Vizváry, M., Ošlejšek, R., Čeleda, P., Tovarňák, D.: Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range. In *2017 IEEE Frontiers in Education Conference*. Indianapolis, IN, USA: IEEE, 2017. p. 1-8, 8 pp.

Contribution (20%): I was responsible for the design of visualizations for cyber defense exercises and wrote several parts of the paper.

Publication type: Conference, IEEE, CORE ranking B.

To cope with the complexity of CDXs and to enable their support in KYPO, we researched and formalized related organizational processes. Our effort resulted in the organization of Cyber Czech – a series of the biggest technical cyber defense exercises in the Czech Republic organized since 2015 in the cooperation with the Czech National Security Agency. Experience and lessons learned are summarized in **Article B**.

The complexity of CDXs is hidden in several aspects that pose high requirements on both computational (cyber ranges) and human resources. It is a team-based exercise with multiple (4-5) teams of learners trying to defend critical infrastructure against a team of attackers simultaneously. Besides teams of learners and attackers, there are teams of technicians responsible for the cyber range management and technical penalizations, people paying the

role of regular users of the critical infrastructure, law enforcement agencies, and judges supervising predefined rules and penalizing teams of learners for their violation, etc. Every team of learners has its own copy of the network, which consists of several zones and tens of nodes.

Although this exercise format is popular and used worldwide by numerous organizers in practice, it has been sparsely researched. In the paper, we contribute to the topic by describing the general exercise life cycle, covering the exercise's development, dry run, execution, evaluation, and repetition (Figure 2.2). Each phase brings many challenges that organizers have to deal with and where a cyber range can support automation. The paper summarizes our lessons learned from the Cyber Czech organization and KYPO development, aiming to help organizers to prepare, run, and repeat successful events systematically.

Article A: Ošlejšek, R., Pitner, T.: Optimization of Cyber Defense Exercises Using Balanced Software Development Methodology. In *International Journal of Information Technologies and Systems Approach.* IGI Global, vol. 14, no. 1, pp. 136–155, 2021.

Contribution (80%): I was the author of idea that cyber defense exercises could be optimized by using balanced software development methodology. I wrote key parts of the paper.

Publication type: Journal, IF 0.12.

Despite the fact that the organization of CDXs in KYPO is possible, it is still a lengthy and expensive process. It takes several months until a new exercise is designed, instantiated in the cyber range, and the training session is conducted. The reason lies in current practice when CDXs are developed ad-hoc and often from scratch. Our recent article **Article A** addresses these issues by the application of a standard software development methodology to the CDX development aiming to formalize organizational processes and shorten the CDX life cycle.

CDXs are in many aspects similar to traditional software projects. It is especially possible to find the parallel between their life cycle and the life cycle of ERP systems – systems that are composed of existing modules that have to be adapted to customer's business processes, deployed at the customer's site, and maintained. In the paper, we put the parallel between ERP systems and cyber ranges. Cyber ranges have to be adjusted for individual customers as well, then instantiated and configured for each CDX.

We utilize standard project management methods to analyze existing CDX life cycles and derive its unified model. The proposed method shows that CDX development has a hybrid character combining both agile and disciplined features that have to be balanced. While introducing elements of agile development could improve the preparation and dry run phases, a balanced approach is required for the evaluation. Moreover, we observed that the whole life cycle is significantly plan-driven.

Chapter 3

Visual Analytics for Cybersecurity

3.1 Visualizations for Learning Analytics

The ability to use visual-based analytical reasoning is essential in many application domains, including biology [132, 129], medicine [137], and urbanization [113]. The positive effects of visual analytics integration into the learning process have also already been identified. The outcomes serve to understand trainees' actions or optimization of the learning environment [232, 142]. As the educational visualization dashboards have gained considerable attention in the field of learning analytics, reviews concerning this matter emerged as well [31, 209]. The visualizations can monitor trainee's progress and help to compare the performance with other peers [93]. They can also increase motivation and encourage trainees to compete or to collaborate [94].

From studies that relate to education and training from a broader view, we can mention a recent survey of Firat and Laramee [84] who introduce a literature classification in the field of interactive visualization for education with a focus on evaluation. They list common categories of educational visualizations from distinct fields. In this respect, our research is unique as it considers more than the educational theory. It also includes the application of hands-on training with practical and technical aspects that are essential to the learning process.

In the cybersecurity research domain, many authors have addressed the challenges related to the design or user evaluation of cybersecurity tools and techniques [220, 26, 17, 72, 8]. They have confirmed the importance of supporting security tasks by visual interfaces. However, these approaches are aimed at the security-related focus only and do not reflect the educational aspect of the training of new experts. Papers that address enhancing computer

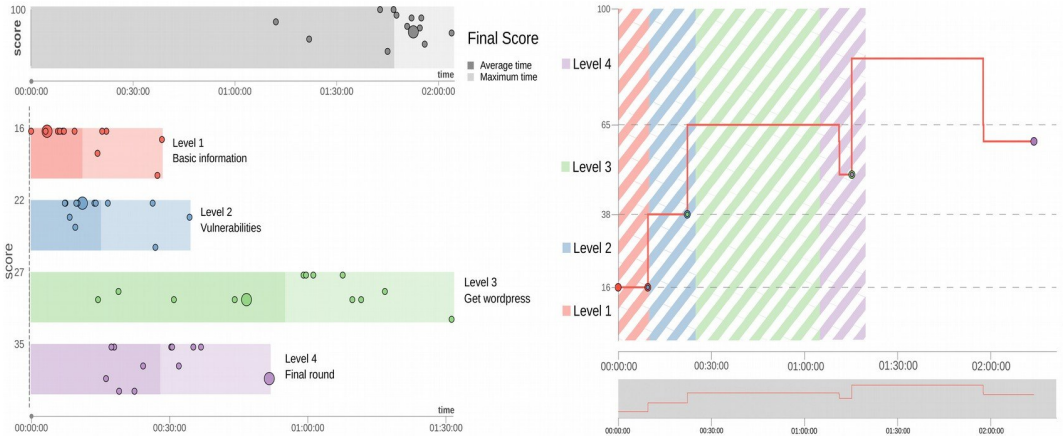

Figure 3.1: Feedback visualizations for players of Capture The Flag games [179].

security skills [208, 259, 85] usually provide outputs of narrow scope and omit the data-driven process-oriented aspects of hands-on cybersecurity training.

We took the existing challenges into account in our research, aiming to incorporate specific features of hands-on cybersecurity exercises into visual-analysis dashboards of the KYPO Cyber Range. Our goal is to provide meaningful insight into cyber processes during training sessions and to thoroughly analyze exercises after their completion so that it is possible to assess the impact on participants and potentially improve future runs. To reach this goal, we build upon respected design frameworks, models, and design methods that exist in the literature [79, 154, 211, 126, 123]. These provide a structure and explanation of activities that designers perform when proposing suitable visualization tools.

Article K: Ošlejšek, R., Vykopal, J., Burská, K., Rusňák, V.: Evaluation of Cyber Defense Exercises Using Visual Analytics Process. In *Proceedings of the 48th IEEE Frontiers in Education Conference (FIE'18)*. San Jose, California, USA: IEEE, 2018. p. 1-9, 9 pp.

Contribution (55%): I'm the author of the idea. I wrote several sections of the paper and supervised the preparation.

Publication type: Conference, IEEE, CORE rank B.

Being aware of the lack of systematic support for the evaluation and post-training analysis of complex cyber defense exercises, we classified and formalized related analytical tasks by applying the Knowledge Generation Model [199] to the domain. The results are discussed in **Article K**.

By using our approach, organizers of CDX events can be systematically supported in their analytical and surveillance activities. Moreover, they could continuously build a knowledge

base that could be shared across organizers in time. In the paper, we formulate three particular analytical goals.

Evaluation of exercise content and parameters reflects the need to make an exercise useful and to keep learners motivated to finish it. Therefore, scenario difficulty, learners' confidence and satisfaction, learners' skills, and many other qualitative aspects should be analyzed.

Behavioral analysis of learners can reveal relevant facts about their motivation, learning impact, or level of knowledge. Gained information is useful for (a) learners as they can learn about themselves, their strengths, weaknesses, and mistakes; (b) exercise contractors, usually learners' employers, who can learn about the skills of their employees; (c) security experts and researchers who can reveal and compare atypical defense strategies, collaboration strategies, and other behavioral patterns.

Runtime situational awareness enables organizers to monitor and analyze the situation on the "battlefield" and actively intervene if necessary. They have to analyze the situation from their perspective and interact with the system continuously.

The paper links these goals to the available cybersecurity data and individual phases of the CDX life cycle, aiming to provide a guideline for the application of visual analytics approaches supporting the analytical goals.

Article E: Ošlejšek, R., Rusňák, V., Dočkalová Burská, K., Švábenský, V., Vykopal, J., Čegan, J.: Conceptual Model of Visual Analytics for Hands-on Cybersecurity Training. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 8, pp. 3425-3437, 2021.

Contribution (30%): I'm the author of the idea and the classification. I wrote several sections of the paper and supervised the preparation.

Publication type: Journal, IF 4.579.

While the previous work has been focused on CDXs, in our recent **Article E**, we map the whole cybersecurity training domain. We use a similar approach, i.e., the adoption of the Knowledge Generation Model. The obtained conceptual model provides a unified life cycle of various training programs. Analytical categories and tasks defined in the paper are described from the perspective of requirements and design decisions. Therefore, the model can serve as a framework for the future development of visual-analytics tools in this filed.

We systematized the visualizations and hypotheses into six categories. Two of them, the insight of trainees and organizing participants, address the run-time visual situational awareness. The remaining four categories are related to post-training visual data analytics: personal feedback, quality of training exercise, behavior analysis, and infrastructure analysis.

Each category is further divided into several sub-categories reflecting user roles and their

finer-grained analytical goals. For each sub-category, we discuss motivation, design rules, and examples of existing visual approaches.

Article J: Vykopal, J., Ošlejšek, R., Burská, K., Zákopčanová, K.: Timely Feedback in Unstructured Cybersecurity Exercises. In *Proceedings of Special Interest Group on Computer Science Education (SIGCSE'18).* USA: ACM, 2018. p. 173-178, 6 pp.

Contribution (30%): I participated in the design of visualization and was responsible for the evaluation. I wrote several sections of the paper.

Publication type: Conference, ACM, CORE rank A.

Besides these conceptual results, we also developed and published several specific visualizations for hands-on training that support trainees in better and faster comprehension of attacks, threats, and defense strategies.

In **Article J**, we investigate how to provide valuable feedback to learners right after a CDX. Based on a scoring system integrated into the cyber range, we have developed a new feedback tool that presents an interactive, personalized timeline of exercise events and helps participants to learn from their experience gained during the exercise. To the best of our knowledge, this was the first paper attempting to study the means of providing visual feedback to learners participating in cyber defense exercises.

In this experimental work, we studied the behavior and interactions of participants at a complex cyber defense exercise. The exercise was focused on defending critical information infrastructure (particularly railway infrastructure administration) against skilled and coordinated attackers. After the exercise, an automatically generated feedback was presented to each team. The feedback application visually encoded a score development. Moreover, learners were able to provide us with their reflection on obtained penalties and awarded points. All interactions with the feedback application, including mouse clicks, mouse movements, and selected options, were logged. This data, together with answers from a short survey, was used to evaluate the usefulness of the timely feedback.

The results show that learners did use the new tool and rated it positively. Since the feedback is not bound to a particular defense exercise, it can be applied to all exercises that employ scoring based on the evaluation of individual exercise objectives. As a result, it enables the learner to immediately reflect on the experience gained.

Article I: Ošlejšek, R., Rusňák, V., Burská, K., Švábenský, V., Vykopal, J.: Visual Feedback for Players of Multi-Level Capture the Flag Games: Field Usability Study. In *IEEE Symposium on Visualization for Cyber Security (VizSec)*. Vancouver, BC, Canada: IEEE, 2019. p. 1-11, 11 pp.

Contribution (30%): I participated in the design of visualization. I wrote several sections of the paper.

Publication type: Conference, IEEE, CORE rank C, MS Academic rating B-.

Article I also focuses on providing feedback to trainees. However, it addresses CTF games whose data differs from Cyber Defense Exercises. In the paper, we describe exploratory feedback visualizations (Figure 3.1) and the results of their user evaluation. In collaboration with domain experts, we classified the visual feedback requirements into three categories that cover trainees' expectations of the training (personalized feedback, comparative feedback, and overall results). Then we applied visualization techniques in the domain of hands-on cybersecurity training in order to provide better insight into the trainees' results right after the training session. We performed a formal evaluation that confirmed the meaningfulness of the defined requirements and usefulness of the post-game analysis visualizations for CTF games.

A conducted user study proved that the tool enables trainees to analyze their results very quickly and compare them with expected behavior or the behavior of other participants. The evaluation also brought several interesting observations. For example, we found out that trainees prefer thr exploration of personal results to the overall game results and comparison with others. Also, our preliminary expectations that the design of the post-training feedback has to be as simple and straightforward as possible have not been confirmed. On the contrary, the trainees used two complementary views, a much simpler clustering preview of the results and the more complex timeline view, with similar intensity.

Article G: Dočkalová Burská, K., Rusňák, V., Ošlejšek, R.: Enhancing situational awareness for tutors of cybersecurity capture the flag games. In *25th International Conference Information Visualisation (IV)*. IEEE, 2021. p. 236–243, 8 pp.

Contribution (33%): I participated in the design and evaluation. I wrote several sections of the paper.

Publication type: Conference, IEEE, CORE rank B.

Analytical dashboard published in **Article G** presents a visual tool intended for tutors of CTF games. As cybersecurity training sessions are process-oriented, tutors have only a limited insight into what trainees are doing and how they deal with the tasks. From their perspective, it is necessary to have situational awareness, enabling them to identify and

react to any issues during a training session as soon as they emerge. We developed a tool that provides educators with timely feedback through the session. More specifically, the tool informs educators of the training progression, helps identify the students who might struggle with their tasks, and reveals overall deviation from the schedule.

The tool has been validated through formative and summative qualitative in-lab evaluations. The participants appraised the impact on the training workflow and gave further insights regarding the tool. In the paper, we discuss the insights and recommendations that arose from the evaluations as they could aid the design of future tools for supporting educators, not only of CTF games but also in other domains.

Although the tool has been designed for on-site training and integrated into the KYPO Cyber Range, it has been used successfully also for the training sessions held remotely due to the COVID-19 pandemic. It would be virtually impossible to organize supervised CTF sessions online without having this analytical dashboard available.

Article F: Dočkalová Burská, K., Rusňák, V., Ošlejšek, R.: Data-driven insight into the puzzle-based cybersecurity training. In *Computers & Graphics*. IEEE, 2022. In press, 11 pp.

Contribution (33%): I was responsible for conceptualization and methodology. I wrote several sections of the paper.

Publication type: Journal, IF 1.936.

Article F summarizes our achievements in post-training analysis of CTF games. The paper describes analytical visualizations intended for tutors and developers of training content. Through a visualization design study, we impelemented a post-training dashboard that supports learning analysis of a single hands-on session. It allows an in-depth trainee comparison and enables the identification of flaws in assignments. The participants apprised the positive influence of the tool on their workflows. Although the dashboard has been designed for cybersecurity CTF games, it is based on more general principles of so-called puzzle-based gamification. Therefore, our insights and recommendations could aid the design of future tools supporting educators, even beyond cyber security.

3.2 Visualizations for Forensic Investigation

Forensic investigation depends heavily on a proper evaluation of collected evidence. Methods of digital forensics [44, 122] are employed for systematic scrutiny of the data. Visual analysis methods are often used to accelerate the investigation and to reveal relationships hidden in the big data.

So far, big attention has been paid to the investigation of network communication [229, 97, 34] and analysis of system logs [115, 114, 221]. However, disks and permanent storage provide another valuable source of information for the digital investigation.

Disk and file systems analysis can be performed in several layers [43]. Approaches addressing specific features are, for example, Change-link 2.0 [139], which provides several visualizations to capture changes to files and directories over time, or the work of Heitzmann et al. [106], who proposed a visual representation of access control permissions in a standard hierarchical file system using treemaps. The utilization of file system metadata for forensic investigation is discussed in [121, 195, 38, 175, 37].

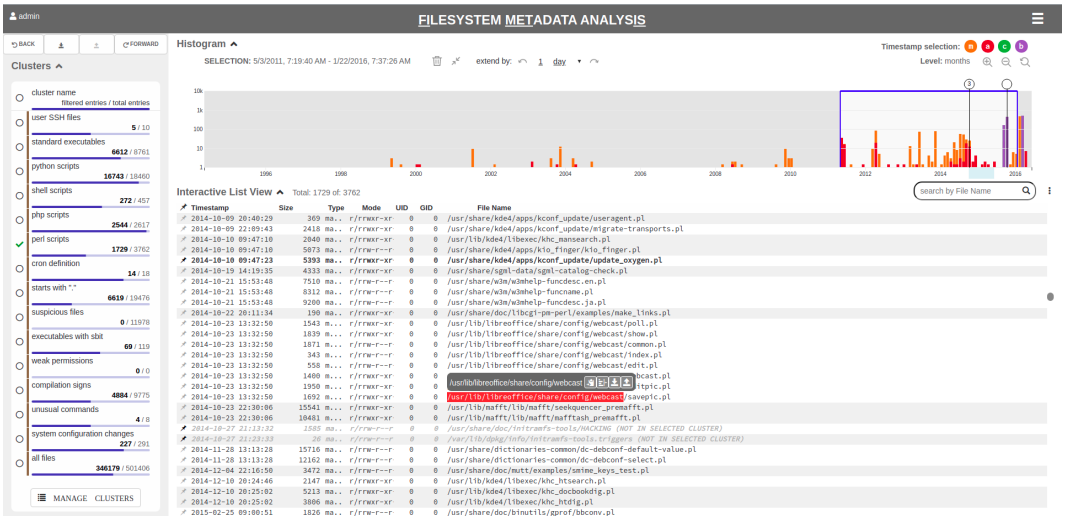

Figure 3.2: A visual-analysis tool for forensic investigation [24].

Cybersecurity training and forensic investigation domains significantly overlap. In both cases, it is necessary to comprehend the complex cybersecurity data and their internal relationships. Therefore, we tackle the forensic analysis using similar user-centered approaches that we also use for the learning analytics. We tightly cooperate with experts who participate in hands-on training programs and who, simultaneously, investigate real incidents.

Article H: Beran, M., Hrdina, F., Kouřil, D., Ošlejšek, R., Zákopčanová, K.: Exploratory Analysis of File System Metadata for Rapid Investigation of Security Incidents. In *IEEE Symposium on Visualization for Cyber Security (VizSec)*. Salt Lake City, US: IEEE, 2020. p. 11–20, 10 pp.

Contribution (20%): I participated in the design of visualization and was responsible for the evaluation. I wrote several sections of the paper. The authors are sorted alphabetically.

Publication type: Conference, IEEE, CORE rank C, MS Academic rating B-.

Our first results achieved in the forensic investigation domain are described in **Article H**. Together with the cybersecurity experts, we identified a gap in providing an intuitive, efficient exploration of file system metadata as part of the cybersecurity incident investigation workflow. We proposed and developed a FIMETIS (FIlesystem METadata analysIS) tool (Figure 3.2) for interactive visual exploration of disk snapshots.

The user interface consists of three coordinated views. A *list view* is a dominant part of the dashboard where the raw file system metadata can be explored. Efficient searching, filtering, block skipping, and bookmarking are supported. A *histogram section* provides an interactive view of data distribution. The view enables per-attribute and span window filtering. A *clusters section* represents a generic mechanism for selecting files or directories with a specific "fingerprint", e.g., files indicating a compilation process.

The conducted user evaluation proved excellent usability and positive impact of the tool on the rapid investigation. All of the analysts were able to provide an incident report at surprising precision very quickly. Moreover, it seems that the results obtained from less and more skilled analysts are subtle. Another interesting observation was made regarding the usage of proposed visual-analytics concepts and their combinations. We noticed different workflows in using the tool by different analysts. This finding indicates that the tool is sufficiently generic. It does not restrict analysts in the investigation strategy. Various approaches to the verification of hypotheses and collecting the evidence can be used. These preliminary results are very promising, and we already work on the extended version that would support the investigation workflow even better.

Chapter 4

Conclusion

In this text, I have presented my research contributions to the progress within the area of cybersecurity research and training platforms and related analytical visualizations. The individual research contributions were accompanied by selected representative articles I co-authored, which are also attached to this text.

In the future, I would like to continue developing visual analysis techniques for cybersecurity education. Especially, increasing the impact of the training courses and providing insight into trainees' behavior is crucial but weakly supported areas. Therefore, our recent research focuses primarily on the utilization of process mining methods for the reconstruction of training walkthroughs and the identification of possible flows in training scenarios. Additionally, cybersecurity experts are looking for new interactive techniques that would enable them to perform forensic analysis efficiently. This challenging and still rather unexplored area presents an application domain with high potential. Our generic KYPO Cyber Range provides us with a great opportunity for this research by enabling the repetition of experiments, systematic collection of data, and analysis.

Part II

Collection of Selected Publications

Chapter 5

List of Publications

This appendix contains the total of eleven recent research papers that were selected as the representatives of my contributions within the studied research field. The papers are divided into two categories reflecting the structure of the thesis. First, articles related to the platform and the content of practical training in cyber security are introduced, followed by papers related to the achievements in visual analysis. In each category, the papers are sorted by publication year from newest to oldest.

Articles Related to Hands-on Cybersecurity Training:

Article A: R. Ošlejšek and T. Pitner. Optimization of cyber defense exercises using balanced software development methodology. *International Journal of Information Technologies and Systems Approach.*, 14(1), 2021

Article B: J. Vykopal, M. Vizváry, R. Ošlejšek, P. Čeleda, and D. Tovarňák. Lessons learned from complex hands-on defence exercises in a cyber range. In *2017 IEEE Frontiers in Education Conference*, pages 1–8, Indianapolis, IN, USA, 2017. IEEE

Article C: J. Vykopal, R. Ošlejšek, P. Čeleda, M. Vizváry, and D. Tovarňák. Kypo cyber range: Design and use cases. In *Proceedings of the 12th International Conference on Software Technologies - Volume 1: ICSOFT*, pages 310–321, Madrid, Spain, 2017. SciTePress

Article D: Z. Eichler, R. Ošlejšek, and D. Toth. Kypo: A tool for collaborative study of cyberattacks in safe cloud environment. In *HCI International 2015: Human Aspects of Information Security, Privacy, and Trust*, pages 190–199, Los Angeles, 2015. Springer International Publishing

Articles Related to Visual Analytics:

Article E: R. Ošlejšek, V. Rusňák, K. Burská, V. Švábenský, J. Vykopal, and J. Čegan. Conceptual model of visual analytics for hands-on cybersecurity training. *IEEE Transactions on Visualization and Computer Graphics*, 27(8):3425–3437, 2021

Article F: K. Dočkalová Burská, V. Rusňák, and R. Ošlejšek. Data-driven insight into the puzzle-based cybersecurity training. *Computers & Graphics*, to appear, 2021

Article G: K. Dočkalová Burská, V. Rusňák, and R. Ošlejšek. Enhancing situational awareness for tutors of cybersecurity capture the flag games. In *25th International Conference Information Visualisation (IV).*, pages 236–243. IEEE Computer Society, 2021

Article H: M. Beran, F. Hrdina, D. Kouřil, R. Ošlejšek, and K. Zákopčanová. Exploratory analysis of file system metadata for rapid investigation of security incidents. In *2020 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 11–20. IEEE Computer Society, 2020

Article I: R. Ošlejšek, V. Rusňák, K. Burská, V. Švábenský, and J. Vykopal. Visual feedback for players of multi-level capture the flag games: Field usability study. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–11. IEEE Computer Society, 2019

Article J: J. Vykopal, R. Ošlejšek, K. Burská, and K. Zákopčanová. Timely feedback in unstructured cybersecurity exercises. In *Proceedings of Special Interest Group on Computer Science Education, Baltimore, Maryland, USA, February 21–24, 2018(SIGCSE'18)*, pages 173–178, Baltimore, Maryland, USA, 2018. ACM

Article K: R. Ošlejšek, J. Vykopal, K. Burská, and V. Rusňák. Evaluation of cyber defense exercises using visual analytics process. In *2018 IEEE Frontiers in Education Conference*, pages 1–9, San Jose, California, USA, 2018. IEEE

Chapter 6

Collection of Articles

Article A

Optimization of Cyber Defense Exercises Using Balanced Software Development Methodology

Radek Ošlejšek[1], Tomáš Pitner[1]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic

IJITSA – Int. Journal of Information Technologies and Systems Approach. 2021, 30 pp.

Abstract

Cyber defense exercises (CDXs) represent an effective way to train cybersecurity experts. However, their development is lengthy and expensive. The reason lies in current practice where the CDX life cycle is not sufficiently mapped and formalized, and then exercises are developed ad-hoc. However, the CDX development shares many aspects with software development, especially with ERP systems. This paper presents a generic CDX development method that has been derived from existing CDX life cycles using the SPEM standard meta-model. The analysis of the method revealed bottlenecks in the CDX development process. Observations made from the analysis and discussed in the paper indicate that the organization of CDXs can be significantly optimized by applying a balanced mixed approach with agile preparation and plan-driven disciplined evaluation.

A.1 Introduction

A shortage of cybersecurity workforce poses a critical danger for current companies and nations [191, 164]. As modern society is exposed to the increasing number of cyber threats, there is a growing need to train new cybersecurity experts.

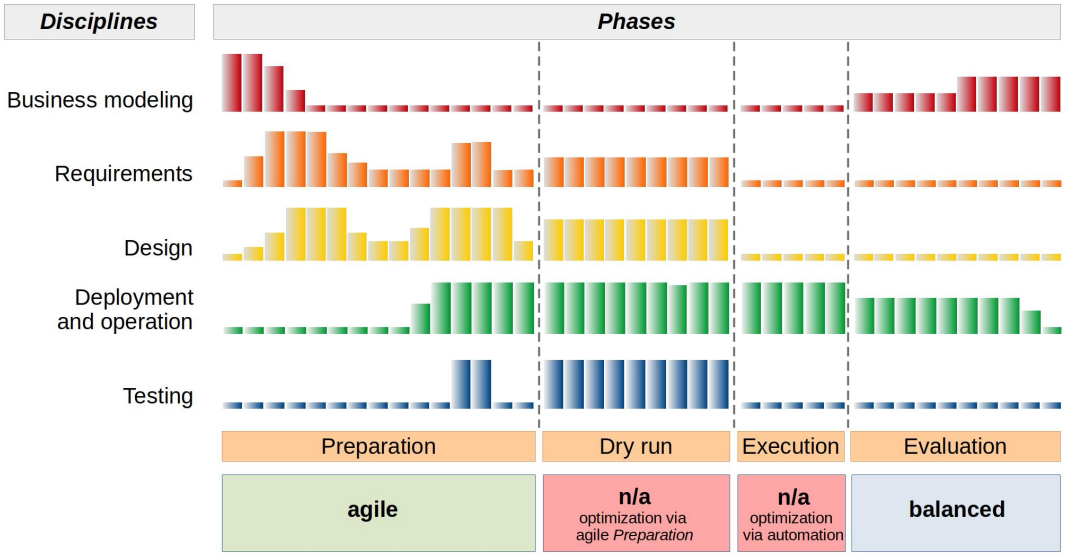

Figure A.0.1: Development method for cyber defense exercises: It consists of four phases and five disciplines. Bar charts suggest approximate work effort required to organize a cyber defense exercise. Observed characteristics of the phases are provided at the bottom of the scheme (*n/a* stands for *not applicable*).

Cyber defense exercises (CDX) [71] represent a popular type of training that aims to fill this skill gap. They have been traditionally organized by military and governmental agencies [185]. CDXs emphasize realistic training scenarios that authentically mimic the operational environment of a real organization [73]. For these reasons, every new CDX is unique. Its preparation requires a considerable amount of skills and workforce. It takes months to prepare and organize a new CDX event with a substantial number of people being involved. These circumstances make the realization of effective hands-on training programs extremely demanding, costly, and with a high risk of failure. One of the reasons is the lack of development methodology when the development of CDXs is rather ad-hoc and loosely driven nowadays.

CDXs are in many aspects similar to traditional software projects. Especially, it is possible to find the parallel between their life cycle and the life cycle of ERP systems [35] – systems which are composed of existing modules that have to be adapted to customer's business processes, deployed at customer's site, and maintained. In the cybersecurity domain, ERP systems are replaced with so-called *cyber ranges*. They represent complex software and hardware environments providing isolated computer networks where cybersecurity exercises can be safely organized without the danger of threatening real users or IT infrastructure. Similarly to the ERP systems that have to be adjusted for individual customers, also cyber ranges have to be adapted, instantiated and configured for each CDX. However, business domains differ. While ERP systems track business resources (e.g., cash, material, or pro-

duction capacity) to support planning, purchasing, and sale, cyber ranges are designed to track vulnerabilities, attacks, network services, and other cybersecurity aspects to support learning processes.

As the parallel between cyber ranges and ERP systems is evident, the utilization of the software development methods for CDX preparation seems to be meaningful. Software companies struggle to optimize the provision of IT services by forcing developers to seek better methods for their business informatics management [237]. In the same way, this paper aims to improve effectiveness and reduce the cost of CDX development by searching for iterative and incremental approaches [136, 19] that could help to deal with the complexity and rapid changes emerging in CDX development and management.

This paper can be seen as a Design Science Research (DSR) with the exaptation type of contribution [98]. Exaptation research extends known solutions to new problems. It is characterized by low maturity of the application domain and high solution maturity. The application domain of this research is the CDX development. As a solution for its low efficiency, the authors aim to use agile or disciplined principles.

This paper contributes to two types of DSR knowledge: prescriptive and descriptive [168]. As a contribution to prescriptive knowledge, a CDX development method is proposed. It is built on the application of the *Software & Systems Process Engineering Metamodel – SPEM* [5] methodology on existing CDX life cycles. This method then serves as a conceptual framework for further analysis. As a contribution to descriptive knowledge, key bottlenecks of the CDX development method are identified. Their reduction using either agile or disciplined principles is discussed.

The remainder of the paper is organized as follows. Section A.2 introduces the related work. In Section A.3, the methodology used for the analysis of existing CDX life cycles is outlined. The CDX development method derived from the analysis and its key features is presented in Section A.4. Section A.5 provides a detailed discussion of the application of agile, disciplined, or balanced methodology on relevant disciplines of the CDX development method. Results and observations are summarized in Section A.6. Section A.7 recapitulates achievements and outlines the direction for future research topics.

A.2 Related Work

The use of agile methodologies has increased significantly over the past decades [66, 109], promoting the value of the human-centric software development process. However, agile development suffers from many limitations [167, 228], and then it is not suitable for all types of projects [90].

On the contrary, traditional plan-driven methods (also call disciplined methods) like Rational Unified Process [133] comes from the assumption that planning and documentation

is the key to successful project management and development. They focus on repeatability, predictability, verification, and validation. However, these features can make plan-driven methods too rigid and hardly adapting to changing requirements.

Balanced (also called hybrid) methods represent a mixture of both the worlds. In [32], the authors provide a comprehensive survey on agile and disciplined methods and discuss the ways of their balancing. They conclude that *"there is no agile or plan-driven method silver bullet"*. Hybrid models combining agile and traditional development can be found in [261, 88]. In [116], the authors show that the hybrid approach should be more scalable than the agile methods and that the hybrid approach can provide better cost-benefit ratios compared to the traditional plan-driven methods.

As the development process of CDXs is ad-hoc and informally driven in current practice, the character of the CDX life cycle is unknown. The authors did not find any work dealing with the application of either agile, disciplined, or balanced methodology on CDX development. Therefore, current knowledge in the field of project management and system development is used to define a CDX development method and discuss its agile vs. disciplined characteristics. This paper builds on the study of existing cyber defense exercises. The literature survey revealed three key papers dealing with organizational aspects of CDXs.

The Cyber Exercise Playbook [125] defines three phases of CDX development and describes user roles participating in the life cycle. The playbook focuses primarily on the planning phase, which is organized as a series of five consecutive meetings. This model is also discussed and summarized in [212].

The CyberRX Playbook [11] introduces four phases. This work emphasizes the need for regular improvement of the cybersecurity program via internal lessons learned. Putting the stress on repeatability and continual improvement puts additional demands on the life cycle.

Probably the most detailed life cycle is discussed in [244]. Their model is based on the experience from the organization of the Cyber Czech exercise. This paper describes the responsibilities of user roles in five phases and also describes significant outputs. Bottlenecks of the development process are discussed, as well. The time and workforce required for the development are identified as critical problems.

Inefficiencies in the CDX life cycle are addressed also in [256]. According to the authors, *"cyber-security exercises are a good tool for cyber-security skill development, but the inefficiencies in cyber-security exercise development and execution life cycle limit its ability to be widely used for cyber-security skill development"*.

Although the core of different life cycles is similar, they vary in many details like the number and names of phases, or names of roles and their responsibilities. They also differ in the level of detail in which the discussion is held. Therefore, it is difficult to generalize them, derive unified concepts, and identify bottlenecks in the workflow that could be eliminated. This paper struggled to fill this gap by providing a unified development method.

Apart from studying CDX life cycles, the researchers also focused on the analysis of existing cyber ranges, as their features can significantly affect CDX development.

The development of cyber ranges has seen a large increase in recent years. There is an extensive survey of state-of-the-art cyber ranges and testbeds in [60].

Although there are many cyber ranges available worldwide, e.g., *Michigan Cyber Range* [156], *SimSpace Cyber Range* [194], or *EDURange* [4], there are not many sources publicly available providing sufficient details about their features and architecture. It is because the cybersecurity domain represents a sensitive area sharing many similarities with military or intelligence services, in which many sources are secret or restricted.

Fortunately, exceptions exist. One very popular cyber range is *DETER/DeterLab* [165, 23], which was started to advance cybersecurity research and education. The description of the architecture, features, and operation can be found also for *CyRIS* [186], *CyTrONE* [27], *NCR* [83], and KYPO [243].

Exploration of these cyber ranges shows that, despite some differences, they share many common features and concepts. This paper primarily builds on the KYPO cyber range platform [243], in whose development are the authors directly involved. However, the presented observations and features related to CDX development can be generalized and valid for all similar modern platforms.

A.3 Research Method

All the conceptual papers that have been found during the literature survey [125, 11, 71, 244] divide the CDX life cycle into several consecutive phases ending with milestones. This fact suggests that the global character of CDXs is rather disciplined.

Based on this observation, SPEM [5] was chosen as a meta-model to be used to analyze the CDX life cycle in detail and provide a methodological view of its development. SPEM can be considered as a continuous evolution of the IBM RUP meta-model [215], where the division of the development into consecutive phases play an important role. This process-oriented meta-model is often used as a baseline framework for the conceptualization of software engineering processes [197, 21, 63].

SPEM provides conceptualization from different perspectives. In this paper, the SPEM is used to comprehend the rationale of the CDX development process and to create a model suitable for the analysis of bottlenecks. This work utilizes the following selected elements of the SPEM 2.0 Base Plug-in [5, p. 155] to get a model with a convenient level of abstraction.

- *Activity kinds:* CDX-specific *phases* and *milestones* from CDX life cycles were defined. Basic *activities* were derived, problems in their implementation were identified. Possible decomposition of phases into iterations was analyzed.

- *Work product kinds:* The main problem of the CDX organization and preparation is low efficiency. Overtaking this bottleneck requires improving the repeatability of the CDX development processes and struggling for its maximal automation. Therefore, during the modeling, attention was payed primarily on *artifacts* that represent tangible elements like documents or formalized knowledge bases.

- *Work product relationship kinds:* Decomposition was omitted to keep the modeling and analysis on a suitable level of abstraction. Instead, this paper deals with a flat model of *dependencies* between *artifacts* (also referred to as *impacted by* relationships in the SPEM meta-model).

- *Category kinds: Roles* were derived from the skills, competencies, and responsibilities of individuals identified in the CDX life cycle. *Disciplines* of SW development, especially of the ERP systems, were identified and adapted to the specifics of the CDX life cycle. Then, the *activities* were further elaborated taking into account *roles*, *work products*, and *work product relationships* involved in them.

The process of the SPEM application was iterative. Section A.4.2 corresponds to the *process structure* perspective of the SPEM [5, p. 43], where *roles* and *phases* are discussed. The *process with methods* perspective [5, p. 95], i.e., the *disciplines*, *artifacts*, and their *dependencies*, is discussed in Section A.5. The authors drew from the existing models of CDX life cycles, but also from the 6-years old experience of the authors with the development of the KYPO cyber range and organization the Cyber Czech CDX. The model was discussed with domain experts – organizers of the Cyber Czech CDX.

The high-level model resulting from the application of the SPEM meta-model on CDX life cycles is shown in Figure A.0.1. The scheme uses two dimensions to capture the approximate effort needed by development activities. The time dimension (the x axis) splits the CDX life cycle into phases, while the workflow dimension (the y axis) includes working activities called disciplines.

Once the conceptual model was available, the research continued in the analysis of the application of agile or disciplined approaches to the critical parts of the CDX development method. This process consisted of two stages. First, problems and possible solutions to the four development phases were identified regardless of discipline. They are described in Section A.4.2. Then, the analysis of activities within individual disciplines and critical phases was conducted. Obtained dependency models were used to formulate recommendations for using agile or disciplined approaches. These per-discipline observations are described in Section A.5 and summarized in Section A.6.

A.4 CDX Development Method

This section focuses primarily on the time dimension of the CDX development method (the x axis in Figure A.0.1). The goal is to describe basic characteristics of phases regardless of disciplines. First, roles involved in phases are introduced.

A.4.1 Roles

Methods of traditional software-system development introduce standard roles for people involved in the process, like analysts, developers, testers, or stakeholders. However, the development of a CDX is specific. It is more similar to adjusting an existing generic system for a particular customer or business domain rather than developing a bright new system from scratch. Archetypal roles defined in this section come from steady terminology established in the field of cybersecurity education where color teams are used to differentiate the responsibilities of people in the exercise [125, 244, 36]. Apart from these "color-named" teams, additional roles are introduced so that the entire CDX life cycle is covered.

Stakeholder

Stakeholders represent an organization whose needs are to be met by the exercise. Cyber defense exercises represent big expensive events that are usually organized on the request of specific customers willing to train their experts. Often, these customers represent bigger commercial subjects operating critical infrastructures, e.g., energy distributors, governmental authorities, ministries, or national security agencies. Stakeholders are always involved in the CDX life cycle. Although the level of their involvement may differ, they often intensively participate in all stages of the CDX life cycle. Stakeholders are usually involved in the content preparation, they are presented as observers during the training event, and they want to be informed about the learning impact on the trainees. On the other hand, some stakeholders perceive CDX as a service and rely on the CDX organization teams that they do the best.

Training Expert

Training experts are skilled in training people. The ultimate goal of any CDX is to train participants properly. However, the impact of the training on participants can be affected by many factors. Training experts are experienced in organizing cybersecurity training sessions. They are able to consider the learning aspects of the exercise. They act as mediators and coordinators between *stakeholders* and IT experts (members of *red*, *white*, and *green teams* – see bellow) aiming to reflect their ideas and expectations in the exercise.

Blue Team

A blue team represents a group of trainees that cooperate during the exercise to defend a computer network against attackers. Blue teams are usually composed of cybersecurity practitioners like network administrators whose motivation is to train and enhance their skills via CDX. Their goal is to secure an entrusted computer network and defend it against attacks of the red team during the CDX training session. A typical CDX event is organized for several (4-5) blue teams, each of which consists of a few (4-5) participants. All blue teams manage identical network infrastructure and face the same attacks of the red team. Members of blue teams do not participate in the development process of CDX. Instead, they can be seen as end-users of the final product.

Red Team

A red team includes people technically skilled and authorized to conduct cybersecurity attacks. During the CDX development, they are responsible for the definition of meaningful attack plans, vulnerabilities, and attack vectors. During the CDX training session, they follow the attack plan to exploit vulnerabilities left in computer networks of blue teams. Based on the success of attacks, the red team assigns penalties to blue teams.

Green Team

A green team includes system operators who are responsible for the cyber range management. Hands-on training sessions are organized in complex underlying infrastructures that have many technical limitations. Knowledge of these technical aspects is necessary during the CDX development to moderate expectations of stakeholders with regard to possibilities of the cyber range. The green team also configures the cyber range for particular CDX. Moreover, members of the team play an important role during the training sessions. They monitor the infrastructure, fixes occasional crashes and infrastructure issues, and revert networks of blue teams to a functional state if they unintentionally cut off the access to the network on their firewall, for instance.

White Team

The goal of CDXs is to train soft skills in addition to technical expertise. A white team, therefore, simulates media requesting reports from blue teams, regular users of defended networks, law enforcement agencies, and other fictitious users that the blue teams have to interact with. Moreover, they act as judges, enforce the rules of the exercise, observe the exercise, score blue teams, and ensure that the competition runs fairly. During the CDX development, they are responsible for the definition of non-technical content of the exercise, like a background story, or tasks of fictitious users with corresponding penalties.

A.4.2 Phases

According to the SPEM standard, a phase represents a significant period in a project, ending with a major management checkpoint, milestone, or set of deliverables. Phases are activities that are not expected to be repeatable during the project life cycle. Every phase can be divided into multiple iterations, as depicted in Figure A.4.1.

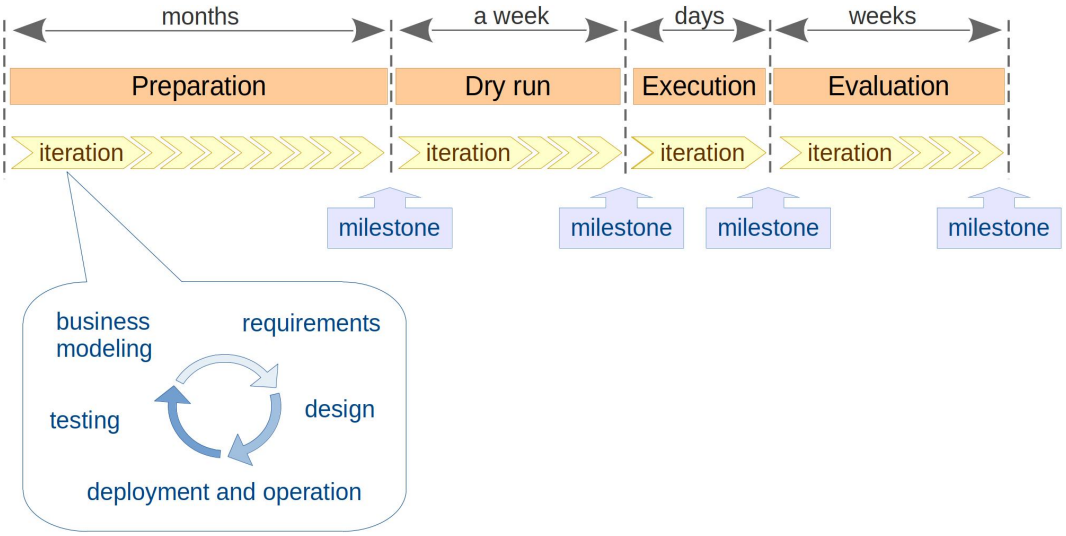

Figure A.4.1: CDX phases and their relation to iterations and milestones.

The CDX development consists of four phases that have been derived from existing CDX life cycles. Their description stays at the conceptual level without going into the details of outputs and activities. These details are described later as part of the discussion of specific disciplines. The text follow the terminology introduced by [244] and [125].

For each phase, a brief description is provided and accompanied by an expected milestone. The milestone captures key achievements that are to be accomplished at the end of each phase. Also, troubles and difficulties related to each phase are summarized. The application of the disciplined or agile approach is discussed as well.

Phase 1: Preparation

Preparation is the first phase of any new CDX. The goal is to define the content of the exercise, specify technical requirements, allocate resources, test coherence of the training scenario, and verify the functionality of the cyber range infrastructure.

Milestone: The cyber range is completely instantiated, configured, and ready for use. Scenarios prescribing the expected steps and tasks of *red* and *white teams* are completed, their coherence and meaningfulness are verified.

Identified troubles: The long-term experience of the authors with developing and organizing CDXs shows that the preparation phase is extremely demanding. It takes several months to prepare a new CDX either from scratch or by significantly changing an existing scenario. Moreover, a lot of people have to be involved in this process and coordinated. These aspects lead to the high rate of errors and logical inconsistencies that have to be revealed and repaired in the later *dry run* phase. These aspects make the preparation phase very expensive.

Solution: CDX preparation is very creative process with unclear requirements at the beginning that have to be clarified by intensive discussion and cooperation of many specialists (*training experts*, *red team*, *white team*, *green team*). Tight cooperation with *stakeholders* and partially with prospective trainees (i.e., *blue teams* participating in the prerequisite testing) is also necessary. Moreover, the budget for the exercise and the schedule of its preparation are usually appointed in advance. These features dominate in agile methods, and then the application of an agile approach to CDX preparation with well-coordinated multiple short iterations should significantly shorten this key phase.

Phase 2: Dry Run

Organizing a CDX is like organizing a mission to the Moon. Every part of the complex infrastructure and all plans have to be well designed and tested before the start. The dry run is similar to beta testing a spacecraft without crew. It follows the same schedule and timing as final exercise to rehearse the entire scenario and interaction between *red*, *white* and *green teams*.

The testing is performed in the same infrastructure that will be used for the final exercise, but without real users (prospective members of *blue teams*). Instead, different people are invited to deputize *blue teams*.

A dry run is conducted even if existing CDX is repeated without changes. It is because cyber range resources are allocated temporarily only for the duration of the exercise and then it is necessary to test it again.

Milestone: The cyber range infrastructure is completely tested and functional. Possible technical issues are fixed. Scenarios of *red* and *white teams* are finalized and orchestrated. Scoring and assessment of *blue teams* are adjusted.

Identified troubles: Although the dry run follows the final training scenario, it takes a much longer time than the real training session due to the reparation of frequent errors and logical inconsistencies.

Solution: Dry run can not be omitted as cyber ranges are too complex, and a CDX represents an event with "the single attempt" when everything has to be working. The reduction of the cost requires the reduction of the frequency of errors so that the dry run could be restricted to only technical testing of unreliable infrastructure. Continuous testing and delivery introduced into the previous *preparation* phase can help to reach this goal. Using the plan-driven CDX life cycle can help. Formalization of artifacts and planing their delivery should enable us to use systems of automated deployment, e.g., Ansible [100]. Also, unit testing can be introduced, which is completely missing in current ad-hoc CDX development. All these steps could make the beta testing substantially less demanding.

Phase 3: Execution

This phase represents the CDX event when real *blue teams* familiarize with the entrusted critical infrastructure, and then they defend it against activities of the *red team*. Simultaneously, they respond to requests of the *white team*. A lot of run-time data is collected during this phase. The data captures activities of all teams, received penalties, etc.

Milestone: The CDX event was realized. Exercise data was collected for further analysis. Hardware resources were released.

Identified troubles: A lot of organizing participants (members of *red*, *green*, and *white teams*) are necessary to organize a single CDX event.

Solution: Automation of tasks. There are attempts to replace the interaction of real people with automated algorithms that are able to follow the training scenario and fulfill the tasks of *red* and *white teams*. The application of either an agile or disciplined approach to the CDX life cycle does not affect this phase.

Phase 4: Evaluation

During the exercise, all participants fully concentrate on their tasks. Especially *blue teams* have only limited awareness of what the *red team* is doing or what were the possible correct reactions to attacks or requests. Therefore, the primary goal of the evaluation phase is to provide feedback to *blue teams* so that they can learn from the exercise. Apart from that, the secondary goal is to retrospectively validate CDX and verify how much it fulfilled expectations of *stakeholders* and *training experts*. In both cases, the run-time data and notes of participants are collected, analyzed, and processed to feedback reports and internal lessons learned.

Milestone: A feedback was prepared and delivered to *blue team* members. Internal lessons learned were formulated and provided to *stakeholders* and *training experts*.

Identified troubles: Nowadays, it takes several weeks to collect and analyze necessary data and to prepare reports and other outputs. It is because the outputs are created informally and ad-hoc. Organizers of a CDX put together their notes, analyze collected data, and together produce desired feedback and internal lessons learned. A lot of manual analysis performed by domain experts is necessary.

Solution: Evaluation is a creative process where people with different expertise have to collaborate tightly. People involved in this process are known in advance. They are *stakeholders*, *training experts*, and members of the *red*, *white*, and *green teams*. Considering these facts, the *evaluation* phase shows the signs of agile development.

On the other hand, the scope of their work is known (i.e., feedback reports and lessons learned), while the time required to prepare the outputs is flexible. Although we attempt to shorten the evaluation and provide feedback as soon as possible, we are aware that the quality of outputs depends on the quality of post-training analysis, which is time demanding. These aspects indicate that introducing a disciplined methodology would be more beneficial.

The traditional triangle *features/scope – resources/cost – schedule/time* used to distinguish between fixed and variable features of methodologies fails, indicating that a balanced approach should be considered. Information gathering should be based on a disciplined approach with formalized artifacts and processes. This formalization enables us to develop supporting tools that would shorten and precise data collection. On the other hand, subsequent agile, iterative creation of feedback and internal lessons learned would support orchestration of involved experts leading in faster outputs.

A.4.3 Summary

Figure A.0.1 summarizes the application of agile or disciplined approaches to individual phases of the CDX life cycle. Using an agile approach to the *preparation* with short iterations, orchestration of people, and continuous testing and delivery of outcomes could significantly shorten this phase and reduce the *dry run* as well. On the contrary, the *evaluation* requires a balanced approach with disciplined information gathering and agile information processing. Neither a disciplined or agile approach has a direct impact on the *execution phase*.

A.5 Detailed Discussion of Disciplines

Disciplines in the software development process represent cross-cutting activities spread over all phases of the life cycle with variable intensity. Since the goal of the CDX development is not related to a cyber range but its content, the five disciplines discussed in this paper lightly differ from what is usually introduced in standard software development.

The goal of this section is to provide a fine-grained view of the character of activities so that the previous observations made during the analysis of phases are proved and explained in more detail. The text focuses primarily on the *preparation* and *evaluation* that appeared to be are relevant for the discussion on the usage of disciplined or agile approaches.

This section is structured as follows. For each discipline, artifacts that represent key tangible outputs are discussed. Then, the responsibilities of individual roles dealing with the artifacts are described, reveal the character of working activities. The approximate work effort required to be spent in various phases is suggested in Figure A.0.1 in the form of bar charts and discussed for each discipline as well. Based on these details, conclusions regarding using either disciplined or agile approaches at the low level of CDX development are formulated.

Artifacts, roles, and responsibilities are also schematically captured in low-level models (see Figure A.5.1, for instance) with the following notation: Responsibilities for the creation of artifacts are captured by horizontal swimlanes with a list of involved roles on the top of each swimlane. Fort he sake of simplicity, activities are omitted. They are only discussed in the text. Instead, dashed arrows are used, representing dependencies (*impacted by* relationships of the SPEM meta-model) between artifacts. Arrows direct from a source artifact (a source of knowledge) to a target artifact (derived knowledge or specification). Artifacts produced by other disciplines are placed out of swimlanes and depicted with less intensive light gray color.

A.5.1 Business Modeling

Business modeling is optional in traditional software-system development. Its goal is to get insight into the business processes of the application domain that should be reflected in the implementation. Often, the business vision and objectives are formulated much earlier than the project is initiated.

In the application domain of this paper, the business is related to hands-on cybersecurity training provided as a service. The business modeling, therefore, corresponds to the knowledge modeling in the field of learning and cyber security. The business view should cover two primary business goals.

First, it is a learning impact. Learning objectives can be derived from the analysis and modeling of existing cybersecurity processes, e.g., attack or cyber-defense strategies [218,

152], so that they reflect new trends and threats.

Second, it is the sustainability of the training program. According to [198], tacit knowledge of domain experts is acquired and shared directly through good quality social interactions and through the development of a transactive memory system. However, CDXs are organized occasionally, and the knowledge gained during the organization of a CDX is lost as people leave the development team. Methods of formal knowledge modeling [29] have to be employed to support long-term knowledge sharing and transfer. Conceptual ideas of knowledge modeling in CDXs can be found in [182], but further research is required in this field. To the best of knowledge of the authors, such pre-training analyses are not conducted in practice due to the missing methodology for CDX development even though they would significantly accelerate exercise preparation.

Artifacts, roles, and responsibility: The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure A.5.1.

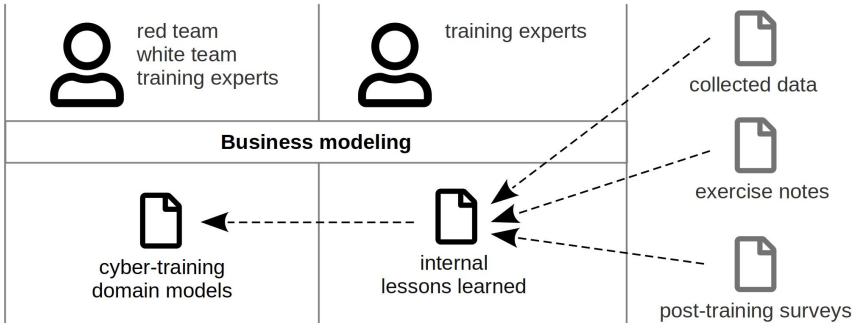

Figure A.5.1: Dependencies between artifacts of the *business modeling* discipline and roles participating in their creation.

- *Cyber-training domain models* – currently, they have the form of informal text documents shared as wiki notes or, more often, they do not exist at all. Most of the knowledge related to the design of the content of CDXs keeps in heads of involved cybersecurity experts, lawyers, and legal experts invited to *red* and *white teams*.

  If formal modeling is introduced in the CDX life cycle, then the *red team* should be responsible for modeling cybersecurity processes, e.g., new vulnerabilities or attack vectors. The *white team* should contribute soft skills to the model, e.g., a classification of low-related objectives. *Training experts* should review the models to be applicable in the educational context.

- *Internal lessons learned* – experience gained from particular exercise and used as supporting material for future exercises and further development of the cyber range. Lessons learned are formulated by *training experts* retrospectively based on the analysis of the *collected data*, *exercise notes*, and *post-exercise surveys* provided by different

people involved in the exercise, as discussed in Section A.5.4. Lessons learned from individual exercises should also be retroactively reflected in the existing *cyber-training domain models*.

Work effort: Business modeling is the most intensive at the beginning of the *preparation* phase, when learning and training objectives are formulated, and during the *evaluation* when lessons learned are derived, and business models are updated according to gained experience.

Disciplined vs. agile character: Elaboration on the *cyber-training domain models* is significantly creative work requiring the collaboration and synchronization of many experts. Therefore, the agile approach in both the *preparation* and *evaluation* phases should be preferred. On the other hand, the formulation of *internal lessons learned* during the *evaluation* requires information structuring, formalization, and well-driven delivery of supporting materials. Otherwise, the outputs are either incomplete or hard to re-use for future exercises. Therefore, a disciplined approach should be preferred in this case.

These observations confirm the agile character of the *preparation* phase and the balanced character of the *evaluation* phase, as discussed in Section A.4.2.

A.5.2 Requirements

Software development distinguishes two types of requirements: functional and non-functional. However, this traditional division fails in the CDX life cycle. Modern cyber ranges are designed as generic, enabling users to organize a wide variety of different exercises through a generic user interface. They are equipped with generic scoring boards, analytical tools, and interfaces providing access to hosts of the defended network, for instance. It is possible to use again the parallel with the ERP system providing a unified interface for variable business goals. Therefore, functional and non-functional requirements can be considered as fixed in this sense. The CDX development methodology deals with exercise development, not cyber range development.

Therefore, the CDX development distinguishes another two requirements: scenario- and infrastructure-related. *Scenario-related requirements* describe the activities of users involved in the exercise. They define what and when the *blue*, *red*, and *white teams* do in the cyber range during the exercise. On the contrary, *infrastructure-related requirements* are linked to the facilities of the cyber range. They include requests put on the configuration of the cyber range, e.g., minimal throughput of network connection.

Artifacts, roles, and responsibility: The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure A.5.2.

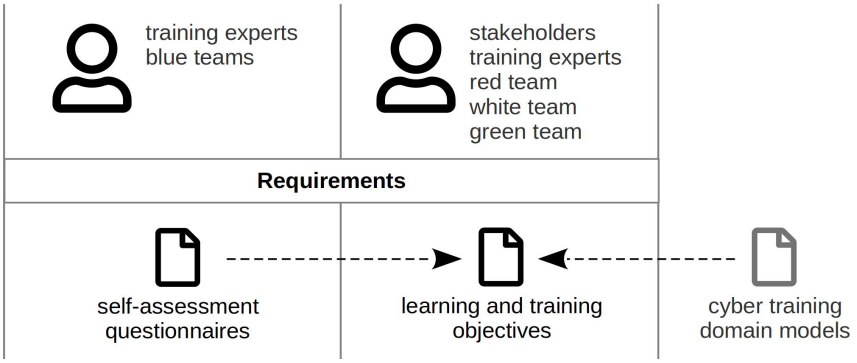

Figure A.5.2: Dependencies between artifacts of the *requirements* discipline and roles participating in their creation.

- *Self-assessment questionnaires* – part of prerequisite testing [239] of *blue team* members. Questionnaires provide insight into the experience and skills of individual trainees. They are defined by *training experts*. Results of self-assessment are used to adjust *learning and training objectives* and for establishing balanced teams.
- *Learning and training objectives* – they define educational requirements that should fit the skills of trainees (*blue team* members) and expectations of *stakeholders*. They are defined by *training experts* together with *stakeholders* and they reflect *self-assessment questionnaires* and *cyber-training domain models*, if available. This artifact includes soft learning objectives as well as requirements put on network topology. *Red team* and *white team* members act as domain experts consulting and reviewing meaningfulness of the objectives from the cyber security and legislation points of view. The *green team* reviews network requirements from the point of view of technical possibilities of the cyber range infrastructure.

  Learning and training objectives can be considered as critical because they form the basis for other artifacts. Improperly selected objectives can lower the impact of the exercise, demotivate trainees to finish the exercise, or demotivate *stakeholders* to further support the training program.

Work effort: Initial requirements are specified during the early stages of the *preparation* phase and then adjusted continuously during this phase. Significant revisions are usually triggered by acceptance tests performed in the *preparation* and *dry run* (see Section A.5.5 for more details). *Self-assessment questionnaires* are usually taken once during the *preparation* phase. However, iterative prerequisite testing would be possible as well.

Disciplined vs. agile character: As *stakeholders* require to train users in new skills, often related to their real critical infrastructures that they operate, CDXs are usually designed from scratch. The *learning and training objectives* that are key in this discipline have

to be invented and defined from the beginning. Their elaboration requires long discussion between *stakeholders* and organizers with short iterations to reach initial definitions as soon as possible. These observations correspond to the agile character of the whole *preparation* phase discussed in Section A.4.2.

A.5.3 Design

The ultimate goal of this process is to think over the details of the exercise, including technical specification being used for the configuration and initialization of the cyber range.

Artifacts, roles, and responsibility: The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure A.5.3.

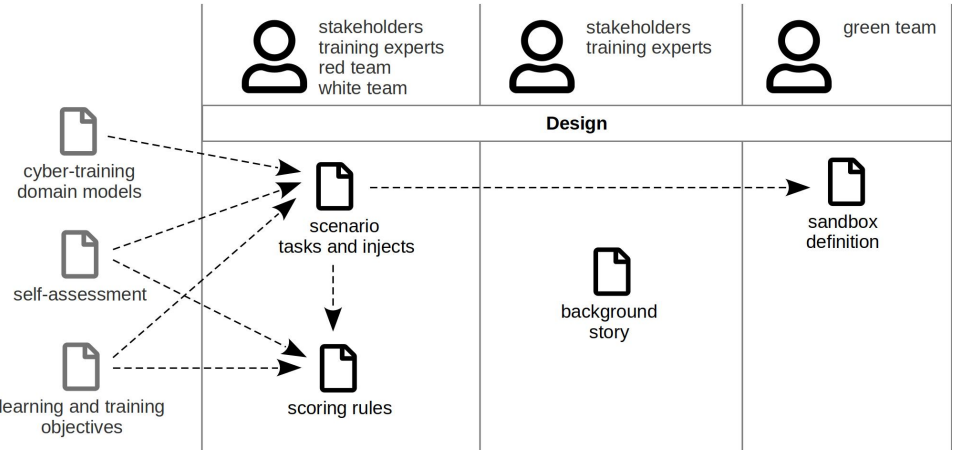

Figure A.5.3: Dependencies between artifacts of the *design* discipline and roles participating in their creation.

- *Scenario tasks and injects* – attack plans of the *red team* and tasks of the *white team* (in the cybersecurity domain, tasks of the *white team* are called injects). Tasks and injects are derived from *learning and training objectives* with respect to the results of *self-assessment questionnaires* and the domain knowledge captured by the *cyber-training domain models*, if available.

  Since the *scenario tasks and injects* artifact is directly linked to the *learning and training objectives*, then also the participating roles are very similar. However, in this process *stakeholders* and *training experts* act as consultants checking whether tasks and injects proposed by *red* and *white team* fits learning and training objectives.

- *Background story* – a fictitious story formulated using the fantasy of *stakeholders* and *training experts* and proving a broader context to *blue teams*. The story explains who

is who in the cyber warfare, what is the organization whose network to be protected, what is the critical infrastructure, and other facts that help *blue teams* understand their goals. Fictitious countries in an escalating conflict are often used to provide trainees with a pseudo-realistic world fallen in the cyber warfare, where a critical infrastructure, e.g., nuclear power plant, has to be protected. This story is later transformed into information sources available to *blue teams* during the exercise, e.g., a news portal, information panels, or oral communication between the *white* and *blue teams*.

- *Sandbox definition* – a structured document capturing the network topology. This technical document is built by the *green team*. It encodes parameters of links and hosts, e.g., throughput, amount of RAM, CPU speed, or IP addresses. It also defines software running on individual hosts. Besides the operating system and applications, it also specifies vulnerabilities that have to be presented on hosts according to the *scenario tasks and injects* artifact. Software to be running on hosts is prepared in the form of disk images that are uploaded on hosts during deployment.

- *Scoring rules* – penalties for unavailability of services, successful attacks of the *red team*, lax or unprofessional response to the requests of the *white team*, technical help of the *green team*, and other possible failures of *blue teams*. Scoring rules are often linked with specific scenario tasks and injects.

  Scoring rules are primarily defined by *stakeholders* and *training experts* who the best understand training and learning objectives. The *red* and *white teams* bring an insight into the difficulty of tasks and injects.

Work effort: During the early stages of the *preparation* phase, a significant effort has to be made to draft tasks, injects, and the background story based on the gradual clarification of learning and training objectives. Another important milestone is a hackathon (see Section A.5.5) during which all the artifacts are finalized and prepared for the first acceptation testing. Artifacts of the design discipline are continuously adjusted after the hackathon and during the acceptance testing.

Disciplined vs. agile character: All the artifacts defined in this discipline are complex and mutually connected. Their concurrent iterative development, together with the artifacts of the *requirements* discipline, is a must. Therefore, the agile approach to their elaboration during the *preparation* phase should be preferred.

A.5.4 Deployment and Operation

In this discipline, organizers configure the cyber range, operate it, and allocate resources. Laboriousness depends on the properties of the cyber range. But in general, these activities

include a lot of continuous manual work.

Artifacts, roles, and responsibility: The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure A.5.4.

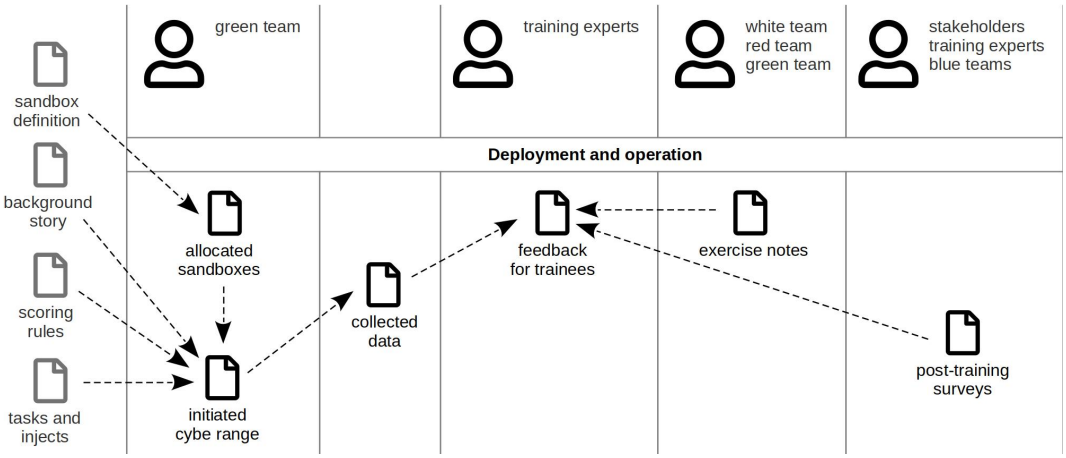

Figure A.5.4: Dependencies between artifacts of the *deployment and operation* discipline and roles participating in their creation.

- *Allocated sandboxes* – an allocated network infrastructure with respect to the *sandbox definition*. The infrastructure can be either emulated in a virtual environment (e.g., in a cloud) or physically wired. The first approach is common in modern cyber ranges. Regardless of the realization, it is always a lengthy, unreliable process. The experience of the researchers shows that even a cloud-based emulation takes long minutes or hours to allocate complex networks of CDXs. Moreover, the allocation often fails for various technical reasons. Manual intervention and continuous testing by members of the *green team* are, therefore, always necessary.

- *Initiated cyber range* – instantiated and properly configured cyber range connected with *allocated sandboxes*. Cyber ranges represent complex software systems consisting of many mutually cooperating components that have to be properly configured and orchestrated. Typical sub-systems that have to be initiated are online user tools, scoring, data monitoring, automated attack generators, traffic generators, etc. The configuration process follows information included in the *background story*, *scoring rules*, and *tasks and injects*. The *green team* is responsible for the cyber range initiation.

- *Collected data* – a run-time data collected during the cyber range operation. The data is monitored and stored automatically by the *initiated cyber range*. The data captured during the *execution* phase and used for detail analysis of the exercise includes, for instance, performed attacks, injects, and their results, command history from individual hosts, or score development.

- *Exercise notes* – experience of *red*, *white*, and *green teams* gained during the exercise.
- *Post-training surveys* – questionnaires capturing the experience of *blue team* members. Surveys are defined by *stakeholders* and *training experts* to reflect their views on learning interests.
- *Feedback for trainees* – results of the analysis of the *collected data* and personal experience of participants. Feedback enables members of the *blue teams* to learn from their behavior and mistakes. It has the form of statistical graphs, notes of *red*, *white*, and *green teams*, and other more or less formal explanations. Feedback is either created manually by *training experts* during the *evaluation* phase or automatically at the end of the *execution* phase.

Work effort: Deployment and configuration activities are the most intensive at the end of the *preparation* phase, and right after the *dry run* when the cyber range is often re-configured, and sandboxes are re-allocated. Operational activities are dominant during the *execution* and *evaluation* phases when the data is collected and analyzed. The cyber range initiation, allocation of resources (sandboxes), and data collection can be significantly automated. The level of automation and continuous delivery is affected by features and possibilities of used cyber range.

Disciplined vs. agile character: As the activities performed during the *preparation* phase include automated processes (cyber range initiation and the allocation of resources), the discussion of the application of either disciplined or agile approaches is irrelevant.

If the gathering of *exercise notes* and *post-training surveys* during the *evaluation* phase is informal, then agile preparation of the *feedback for trainees* would be used to deliver relevant information in a reasonable time. On the other hand, if the gathering of these artifacts is disciplined with predefined structure and deadlines, then the preparation of the feedback would be the matter of fast one-shot analysis. However, structuring the data is not that simple. It is possible to derive and classify common features of cybersecurity exercises, but the content and the realization of exercises differ. Therefore, it is necessary to support the gathering of unexpected informal pieces of information because they often provide very relevant and valuable pieces of information. A balanced approach to feedback preparation is, therefore, required.

A.5.5 Testing

Although the primary concerns of any CDX are related to learning impact, it is virtually impossible to test learning objectives and exercise difficulty. Organizers cannot reveal the content of the exercise to real trainees in advance to check its features. And tests conducted with dummy trainees are confusing due to their different skills and experience. Therefore,

testing is restricted only to the verification of technical aspects and logical consistency of tasks and injects. In current practice, it is organized as two separate events dealing with two levels of acceptance testing.

A *hackathon* is equivalent to alpha testing. *Scenario tasks and injects* and *sandbox definitions* are evaluated by organizers in an intensive full-day workshop at the end of the *preparation* phase.

The *dry run* is equivalent to beta testing. Its goal is to verify the proposed cyber exercise completely and to get diverse feedback on it. The training session is conducted with dummy users, and then also this test can verify only technical aspects of the exercise, not educational. Since the *dry run* represents a separate phase that has been already discussed in Section A.4.2, it is omitted from further discussion in this section.

Artifacts, roles, and responsibility: *Hackathon* is organized by *red*, *white*, and *green teams*. *Dry run*, in addition, involves *blue teams* but consisting of dummy trainees. During the acceptance testing sessions, observed flows are immediately repaired by revising artifacts discussed above. No new artifacts are created.

Work effort: *Hackathon* is organized at the end of the *preparation* phase, followed by a short period of quick fixes of discovered errors. *Dry run* is in the CDX process model captured by a separate intensive phase.

Disciplined vs. agile character: As the current practice in CDX testing is concentrated on two special events, these events increase time and cost. The best practices of agile development require ensuring the quality of the software product throughout the development process. Techniques of continuous testing and deployment are used to test early and often inside short iterations. Therefore, the already discussed agile approach to the *preparation* phase could reduce alpha testing and possibly eliminate the *hackathon*.

A.6 Summary and Lessons Learned

This section summarizes observations made on the application of either agile, disciplined, or balanced approaches to CDX development.

CDX life cycle is plan-driven. The analysis of existing CDX life cycles revealed strong evidence of the plan-driven approach, similar to the RUP process framework, for instance. The life cycles consist of several well-defined phases, each specifying exact milestones, responsibilities, and artifacts. However, artifacts and development processes are often informal and ad-hoc in current practice. To introduce a real plan-driven methodology, their

formalization and putting stress on their precise documentation is necessary. It can bring many benefits. Well-documented artifacts can be re-used in future CDXs. If they are well-structured, then they can also be used for the automation of selected processes. For example, the cyber range would be able to allocate complex sandboxes without the manual intervention of technicians automatically. Plan-driven development also brings better planning and management with verifiable deadlines and outputs. All these features contribute to the acceleration of the organization of CDX programs and their cost reduction.

Table A.6.1: Identified characteristics of individual disciplines; *n/a* = not applicable.

	preparation	**evaluation**
business modeling	agile	balanced
requirements	agile	n/a
design	agile	n/a
deployment and operation	n/a	balanced
testing	agile	n/a

The proposed unified CDX development method, which is based on the analysis of existing CDX life cycles, introduces four phases. Analysis of these phases revealed further details about their features that are summarized in Table A.6.1 and discussed in what follows.

The *preparation* phase is agile. The *preparation* phase shows the signs of agile development. This observation was proved by the detailed analysis of individual disciplines comprising of business modeling, requirements analysis, design, deployment & operation, and testing. Except for the deployment & operation, which turned out to be irrelevant, the application of agile approaches to other disciplines could significantly reduce the time and cost of this phase.

The *evaluation* phase is balanced. The analysis revealed that the relevant disciplines of the *evaluation* phase are business modeling and deployment & operation. They show signs of both agile and disciplined features, making a balanced approach best suitable for the optimization of this phase.

The *dry run* and *execution* phases are not relevant. Applying either agile or disciplined approaches to these phases does not make sense due to the nature of corresponding activities. However, their cost can be reduced by the already discussed introduction of the plan-driven methodology into the whole CDX life cycle and agile approach to the *preparation* phase.

A.7 Conclusion and Future Work

Hands-on cyber defense exercises are crucial in educating the future workforce. However, their preparation is complex, then lengthy, and expensive. This research utilized standard methods of project management to analyze existing CDX life cycles and to derive its unified model. The proposed method shows that CDX development has a hybrid character combining both agile and disciplined features that have to be balanced. While introducing elements of agile development could improve the *preparation* and *dry run* phases, a balanced approach is required for the *evaluation*. Moreover, the whole life cycle is significantly plan-driven.

The main limitation of the presented research is the conceptual level of results. This paper provides a conceptual view and generic discussion. The authors believe that even the gradual adoption of recommendations based on the observations presented in this paper can significantly reduce the cost of CDX preparation, making this kind of cybersecurity training more sustainable, available, and efficient. However, it is a long-term process that requires additional research elaborating on how the adoption should be realized in practice in detail. Introduced development method, together with observations made from the model, can serve as a framework for further investigation.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" project granted by the Ministry of Education, Youth and Sports of the Czech Republic under No. CZ.02.1.01/0.0/0.0/16_019/0000822.

Article B

Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range

Jan Vykopal[1], Martin Vizváry[1], Radek Ošlejšek[2], Pavel Čeleda[1], Daniel Tovarňák[1]

[1] Masaryk University, Institute of Computer Science, Brno, Czech Republic
[2] Masaryk University, Faculty of Informatics, Brno, Czech Republic

FIE – IEEE Frontiers in Education Conference. IEEE, 2017, p. 1-8, 8 pp.

Abstract

We need more skilled cybersecurity professionals because the number of cyber threats and ingenuity of attackers is ever growing. Knowledge and skills required for cyber defence can be developed and exercised by lectures and lab sessions, or by active learning, which is seen as a promising and attractive alternative. In this paper, we present experience gained from the preparation and execution of cyber defence exercises involving various participants in a cyber range. The exercises follow a Red vs. Blue team format, in which the Red team conducts malicious activities against emulated networks and systems that have to be defended by Blue teams of learners. Although this exercise format is popular and used worldwide by numerous organizers in practice, it has been sparsely researched. We contribute to the topic by describing the general exercise life cycle, covering the exercise's development, dry run, execution, evaluation, and repetition. Each phase brings several challenges that exercise organizers have to deal with. We present lessons learned that can help organizers to prepare, run and repeat successful events systematically, with lower effort and costs, and avoid a trial-and-error approach that is often used.

B.1 Introduction

Information and communication systems are exposed to an increasing number of attacks. Apart from simple attacks conducted by hacktivists and inexperienced individuals that can be tracked down [187], there are professional teams backed by organized crime groups or even governments [148] that carefully hide their activities. A shortage in cyber security skills and cyber security professionals is a critical vulnerability for companies and nations [191, 54].

Cyber security can be taught not only using conventional methods, including classroom lectures, seminars or home assignments, but also by hands-on experience. In recent years, there has been a significant growth of hands-on competitions, challenges, and exercises [59, 170]. It is believed that they enable participants to effectively gain or practise diverse cyber security skills in an attractive way.

The most popular events are Capture The Flag (CTF) games [59] and Cyber Defence eXercises (CDX) [170]. While CTF games focus on attacking, defending or both, CDXs train solely the defence. CTFs which put participants in the role of the attacker support the development of adversarial thinking that is necessary for anticipating future offensive actions [166]. CDXs enable participants to experience cyber attacks first-hand.

Although both types of events are prepared and carried out by numerous sponsors for a large number of participants, there are only a few public research papers dealing with the design of an exercise in a cyber range. Granåsen and Andersson conducted a case study on measuring team effectiveness in Baltic Cyber Shield 2010, a multi-national civil-military CDX [95]. They described the instrumentation and collection of data from the exercise's infrastructure and participants in order to provide situational awareness for organizers during the exercise. The Spanish National Cybersecurity Institute proposed a taxonomy of cyber exercises [71] which recognizes operations-based exercises focused on incident response by participants in technical and management roles. ISO/TC 223 effort resulted into ISO 22398, which describe general guidelines for exercises including basic terms and definitions [117]. Unfortunately, technical implementation details of an exercise in a cyber range is out of scope of this standard.

In our work, we address the gap in the literature by describing the life cycle of a complex cyber defence exercise and challenges related to the exercise's design, development, execution and repeatability. This knowledge is based on our experience gained by developing and delivering six runs of a cyber defence exercise scenario with about 120 national and international learners between 2015 and 2017. The exercises have been carried out in a cyber range we are developing and continuously enhancing in order to suit an exercise's requirements.

This paper is organized into five sections. Section 2 provides an overview of existing platforms that can be used as a vehicle for cyber exercises. Section 3 describes a cyber defence exercise carried out in a cyber range. Section 4 reports on lessons learned through six runs of this exercise. Finally, Section 5 concludes the paper and outlines future work.

B.2 Hands-on learning environments

In this section, we give a brief overview of learning environments that can be used in active learning of cyber security. We have done a systematic literature review from 2013 to 2017 to cover recent advances and innovations.

B.2.1 Generic testbeds

Generic testbeds provide a basic functionality for the emulation of computer networks. *Emulab/Netbed* [252] is a cluster testbed providing services for the deployment of virtual appliances, configuration of flexible network topologies and emulation of various network characteristics. Emulab allocates computing resources for a specified network and instantiates it at a dedicated hardware infrastructure. *CyberVAN* [3] experimentation testbed provides a virtualized environment where arbitrary applications running on Xen-based virtual machines can be interconnected by arbitrary network topologies. It employs network simulators such as OPNET, QualNet, ns-2, or ns-3, so the network traffic of emulated hosts travels through the simulated network. This hybrid emulation enables the simulation of large strategic networks approximating a large ISP network.

B.2.2 Lightweight platforms

Several lightweight platforms have been developed for cyber security training. While some of them evolved from the generic testbeds, others were designed from scratch with different needs in mind. *Avatao* [39, 1] is a web-based online e-learning platform offering IT security challenges (hands-on exercises), which can be organized to a path which leads to fulfilling an ultimate learning objective. *CTF365* [56] (Capture The Flag 365) is a training platform that leverages gamification to improve retention rate and speed up the learning and training curve. In the *Hacking-Lab* [210] online platform, teams of participants have to perform several tasks simultaneously; keep applications up and running, find and patch vulnerabilities, solve challenges and attack their competitors' applications.The *iCTF* framework [236] has been developed at The University of California for hosting their iCTF, the largest capture the flag competition in the world. *InCTF* [189] is a modification of the iCTF framework. Using Docker containers instead of virtual machines enhances the overall game experience and simplifies the organization of attack-defence competitions for a larger number of participants.

B.2.3 Cyber ranges

Cyber ranges represent complex virtual environments that are used not only for cyberwarfare training, but also for cybertechnology development, forensic analysis and other cyber-related

issues. There is an extensive survey of state-of-the-art cyber ranges and testbeds [60]. One very popular cyber range is *DETER/DeterLab* [165, 23], which is based on Emulab and was started with the goal of advancing cyber security research and education in 2004. Nowadays, there exist many other cyber ranges, e. g., *National Cyber Range (NCR)* [83], *Michigan Cyber Range (MCR)* [156], *SimSpace Cyber Range* [194], *EDURange* [4], or *KYPO Cyber Range* [243].

B.3 Cyber defence exercise

We have designed a one day Red vs. Blue cyber defence exercise for 50 participants. It was inspired by the Locked Shield exercise [170] organized by NATO Cooperative Cyber Defence Centre of Excellence in Tallinn. We named our exercise Cyber Czech and it has been executed six times so far (2015–2017). Cyber Czech is a hands-on exercise improving the technical and soft skills of security professionals grouped in six Blue teams. It requires substantial preparation effort from the organizers and a dedicated cyber range infrastructure. The exercise involves:

- cloud-based exercise infrastructure (sandboxes),
- training objectives, story, and an exercise scenario,
- participants grouped in teams (Red, Blue, White and Green),
- a physical cyber range facility hosting all participants.

This section explains the cyber defence exercise's components, terms used and definitions, we will use throughout the rest of the paper.

B.3.1 Cyber range infrastructure

The technical part of the cyber exercise relies on a cyber range itself and supportive infrastructure for communication within the exercise and the evaluation of participants' actions. The cyber range emulates a complex network setup in a contained environment. Therefore, participants can realistically interact with an assigned host or network infrastructure, and their actions cannot interfere with the operational environment. The following text describes a high-level view of the architecture of the KYPO [243] cyber range, which we use in the Cyber Czech exercise.

Sandboxes represent a low-level layer of the cyber range. They encapsulate isolated computer networks where users can safely perform their cyber security tasks. Sandboxes are based on virtual appliances placed in a cloud, which makes their allocation, replication,

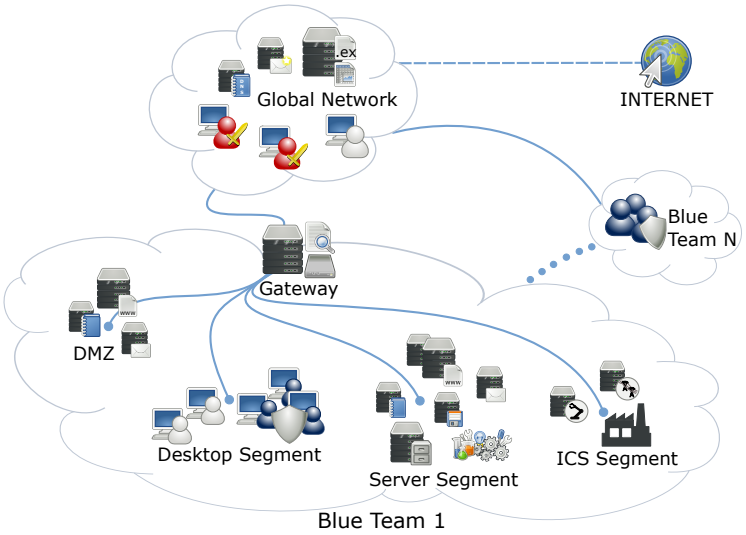

Figure B.3.1: The scheme of the cyber exercise network.

and maintenance easy. Despite the virtualization, neither users nor running applications can recognize that they do not run on a real network.

The scheme of the cyber exercise network is depicted in Figure B.3.1. This network serves as a virtual battlefield with approximately 110 interconnected hosts and other network facilities. It is divided into two subnetworks: *i)* a global network hosting attackers and common network infrastructure, such as DNS and e-mail servers; this network simulates the global Internet, and *ii)* the networks of Blue teams representing the defended network with critical and vulnerable services. Networks of Blue teams are further divided into a demilitarized zone (DMZ), desktops, servers, and industrial control systems (ICS).

Cyber range built-in *monitoring services* cover network traffic statistics, flow data, and full-packet capture. In addition to these off-the-shelf data monitoring services, learners may install their own monitoring applications as a part of their activities inside their sandbox.

Next, we use a generic *logging infrastructure* integrated into the monitoring services. Each host is configured to forward log messages to the central logging server. A processing chain of additional tools is deployed in order to provide real-time access to the normalized log data from the exercise infrastructure. The state of the host's network services is periodically checked and events related to service state changes are logged into the central logging server.

The logging infrastructure is used by a *scoring system* that has been developed to provide feedback to participants during exercise. Penalty points are either computed automatically from events processed by the logging infrastructure (e. g., penalty for inaccessible services) or entered manually. A total score can be shown to participants in real-time. Monitored and logged data is an invaluable input for exercise management, evaluation and further research.

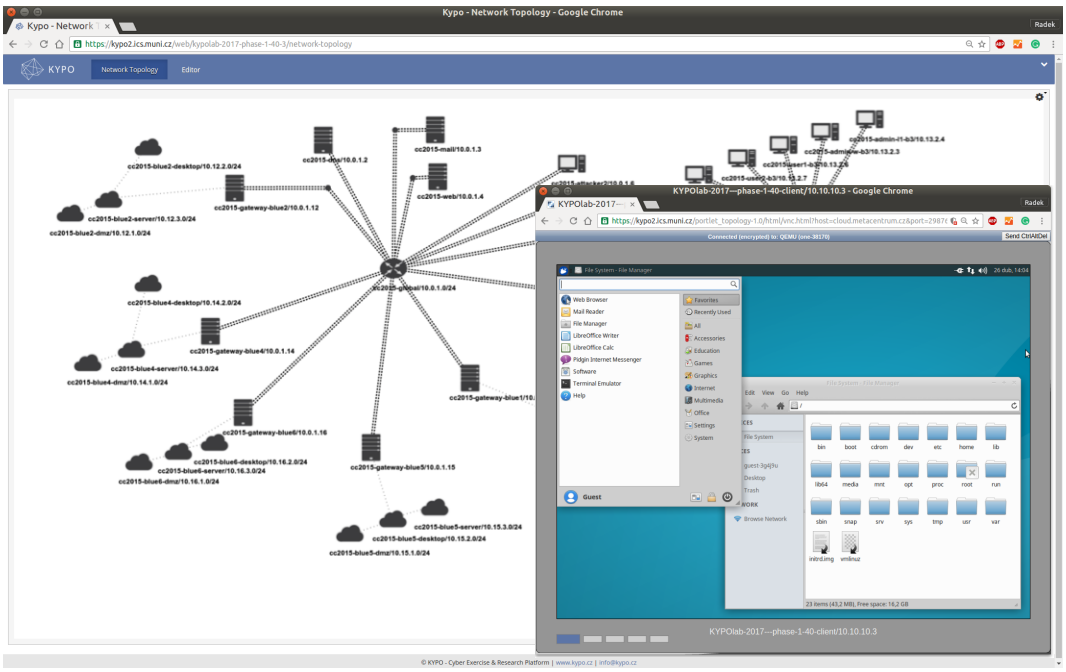

Figure B.3.2: The topology of the exercise network, as seen by participants in the front-end application, and open remote desktop connection to selected host in the separate window.

The *front-end application* provides a web-based user interface to interact with the cyber range. The web interface supports the design and management of sandboxes, single sign-on, remote desktop access etc. We have designed complex interactive visualizations to provide real-time feedback to participants, to provide insights into adversary behaviour, and to build effective situational awareness. Figure B.3.2 shows a screenshot of a sandbox from the Cyber Czech exercise, as was seen by participants in the front-end application.

B.3.2 Exercise objectives, story and scenario

The designed exercise is focused on defending critical information infrastructure against skilled and coordinated attackers. Similarly to other defence exercises, learners are put into the role of members of emergency security teams which are sent into organizations to recover compromised networks. They have to secure the IT infrastructure, investigate possible data exfiltration and collaborate with other emergency teams, the coordinator of the operation and media representatives.

Learners are provided with a background story to introduce them to the situation before they enter the compromised networks. This is very important since the exercise is not set in a real environment and learners have no previous knowledge who is who in the fictitious scenario (e. g., users in their organization, popular news portal, superordinate security team).

They are also provided with technical facts related to the exercise network: network topology including "their" network that will be defended, network architecture and current setup, and access credentials, etc. Before the actual exercise, learners access their emulated network for several hours to get familiar with the exercise. The exercise is driven by a scenario which includes the actions of attackers and assignments for defenders prepared by the organizers. The attackers exploit specific vulnerabilities left in the compromised network in a fixed order which follows a common life cycle in the critical information infrastructure (see Figure B.3.3).

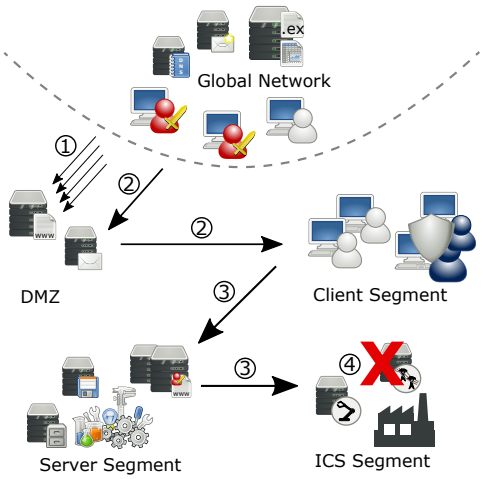

Figure B.3.3: Common attack phases: ① reconnaissance the victim's network; ② exploitation of the unveiled vulnerabilities; ③ escalation of privileges on compromised computers and further exploitation; ④ completing attackers' mission, e. g., shutdown a control system.

The first attack phase involves reconnaissance (scanning of active systems or open network ports). Next, the attackers try to gain access to the machines providing public services (exploitation phase). This is followed by multiple escalations of privileges (accessing segments with internal machines), which enables the completing attackers' mission (shutdown of a critical application). The attackers use a mix of recent and ubiquitous attacks/vulnerabilities that are public and well-known. This is complemented by special tailored malware samples which emulate sophisticated attacks. The completion of each successful attack is recorded by the attackers. On top of that, learners should also answer media requests. The performance of each learners' team is scored based on successful attacks or their mitigation, the availability of specified critical services and the quality of reporting and communication.

B.3.3 Participant roles

Participants are divided into four groups according to their skills, role, and tasks in the exercise. These are now listed according to those commonly used in other cyber exercises:

- *Green team* – a group of operators responsible for the exercise infrastructure (the sandbox in this case). They configure all virtual computers and networks, monitoring and scoring infrastructure. The Green team also monitors the sandbox's health and fixes crashes and infrastructure issues if needed.
- *White team* – exercise managers, referees, organizers, and instructors. They provide the background story, exercise rules and framework for the Red team and Blue teams' competition. The White team assigns tasks (called injects) to the Blue teams and thus simulates media, the operation coordinator, and law enforcement agencies. They might also act as instructors and provide basic hints to Blue teams if needed.
- *Red team* – plays the role of attackers and consists of cyber security professionals. They do not attack targets in the infrastructure of a Blue team randomly, but carefully follow a predefined attack scenario to equally load the Blue teams. This means the Red team exploits vulnerabilities left in a Blue team's network. They should not use any other arbitrary means of attack against the Blue teams. They are also not allowed to attack the service infrastructure. Based on the success of attacks, the Red team assigns penalties to Blue teams. Penalties are assigned manually via a web interface since the amount of awarded points is based on non-trivial factors that need expert review.
- *Blue team* – learners responsible for securing compromised networks and dealing with the Red team's attacks. They have to follow the exercise's rules and local cyber law. The learners are grouped in several Blue teams.

Interactions between the four groups of participants are depicted in Figure B.3.4.

B.4 Lessons learned

Cyber exercises last several hours or days but their preparation typically takes several months involving experts from various fields – IT administrators, penetration testers, incident handlers, managers, legal experts etc. The exercise life cycle consists of several phases that can be mapped to a *Plan–Do–Check–Adjust* (PDCA) cycle. Carefully planning and considering the relationship of all phases may save a significant amount of invested effort and costs. Figure B.4.1 shows the involvement of all teams and effort spent through the cyber exercise life cycle.

B.4.1 Preparation

The preparation phase consumes the majority of work effort and time. First, we have to set the learning and training objectives of the exercise; elaborate the background story and

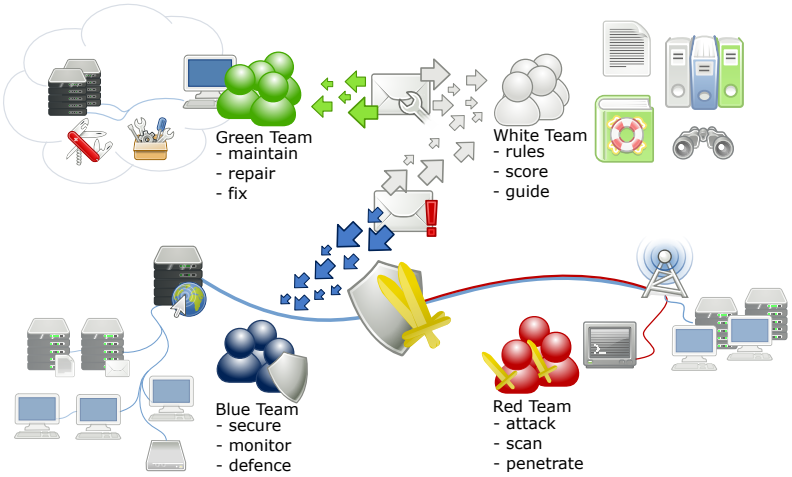

Figure B.3.4: Exercise participants, their interactions and tasks.

develop an exercise scenario consisting of tasks and injects for the Red team and White team – including end users, media and legal representatives. An outline of the exercise scenario is then used for preparing network infrastructure that will be defended by the Blue teams. A more detailed scenario is then used for setting up scoring components: their general weights (e. g., service availability vs. successful attacks vs. reporting) and score structure for every particular service, attack, or inject (e. g., if the Red team is successful in a given attack, the Blue team will be penalized by an exact number of points; if the Red team was successful only partially, the Blue team will be penalized only by a portion of the amount of full points).

In parallel, learners are invited and asked for self-assessment of their skills relevant to the exercise. Based on their input, the White team starts to create Blue teams with balanced skills and experience. The described steps so far correspond to the *Plan* and *Do* phases in the PDCA cycle.

Once the network infrastructure and hosts are configured according to the proposed scenario, they are deployed in a cyber range sandbox. Tasks and injects of the scenario are tested by members of Red team and White team in an intensive full day workshop (hackathon). This is without the presence of Blue teams. The hackathon represents the *Do* and the *Check* phases of the PDCA cycle. After that, there is the last chance to modify the scenario and configuration of exercise infrastructure (the *Adjust* phase).

In our experience, the most challenging tasks in the preparation phase are:

- *Setting learning objectives with respect to the expected readiness of prospective learners*

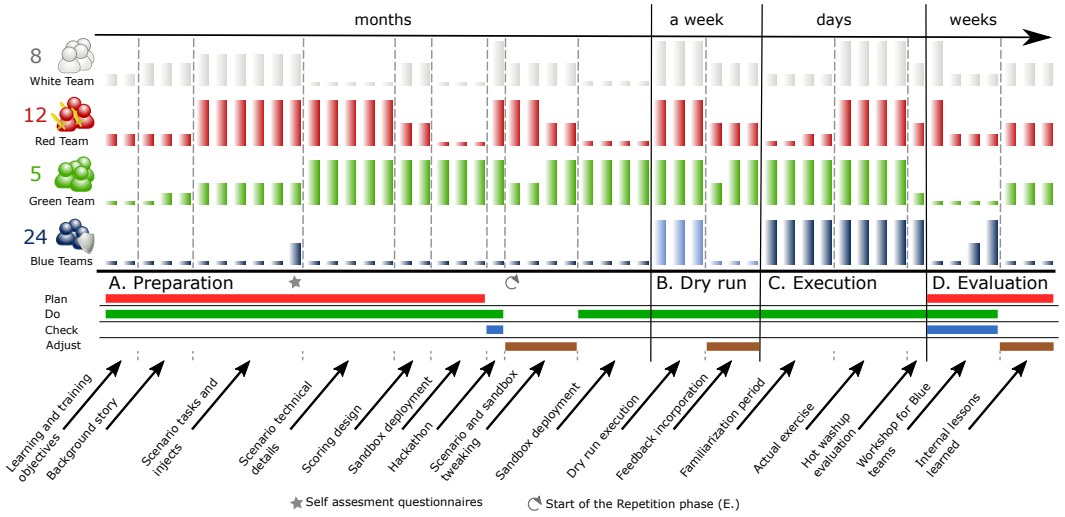

Figure B.4.1: Cyber exercise life cycle in time. Coloured bars show relative effort spent by members of White, Red, Green and Blue teams in respective phases of the life cycle. The four numbers on the left express the size of particular team in the exercise. The mapping to the PDCA cycle is depicted by coloured lines below the life cycle phases.

– the organizers have limited information about learners' skills before the actual exercise. This is a completely different situation to a typical higher education where learners' readiness can be determined by the portfolio of courses passed by the learners. We strongly recommend considering a profile of the prospective learners in order to balance learning objectives and learners' proficiency. The self-assessment questionnaires may provide useful information. The key success factor is to ask questions which are relevant to particular skills that will be exercised, e. g., *What tools do you use for detecting cyber attacks?* instead of *What is your experience with the detection of cyber attacks?*

- *Creating balanced teams* – one of the main aspects of the exercise is to build a sense of teamwork. We advise paying a large amount of attention to creating teams of learners who possess the necessary skills. For instance, if the self-assessment inputs indicate that some learners are experts in one area, it is recommended to distribute them to all teams equally and complement them with experts in another area.

- *Sandbox configuration documents* – continually editing and updating the specification of used systems, network configurations and vulnerabilities is crucial for the successful and smooth preparation of the sandbox. The description should be done using an automation tool such as Ansible [190] to assure its long-term maintainability. Any static documentation (e. g., a wiki page, readme file) is error prone, and becomes outdated very soon.

B.4.2 Dry run

The dry run is a complete test of the proposed cyber exercise to get diverse feedback on it. We invite different groups of learners (testing Blue teams) that participate in a pilot exercise. Dry run follows the same schedule and timing as final exercise to rehearse the entire scenario and interaction between Red, White and Green teams, even though it consumes a considerable amount of manpower. It is a mix of *Do* and *Adjust* PDCA phases.

We learned that adjusting the scoring system based on the dry run might be misleading if the expertise and size of the Blue teams participating in the dry run is not similar to learners. The progress of the dry run may be also influenced by various exercise conditions and events that may not happen in the final execution.

B.4.3 Execution

The execution phase starts with a familiarization period that enables Blue teams to learn about the exercise infrastructure that has to be defended. The Red team takes no action in this period, so the Blue teams have an opportunity to harden "their" infrastructure. Then the actual exercise starts according to the scenario that is strictly followed by members of the Red and White teams. Once the exercise ends, representatives of the Red, White and Green teams provide a very short assessment of Blue teams' performance during the whole exercise (hot wash-up). This is very desirable since Blue teams can see the final score and can only estimate the content of the exercise scenario.

We identified five challenges related to the execution phase:

- *The level of guidance by organizers* – although creating balanced teams should help to equate learners' proficiency and exercise difficulty, the learners sometimes struggle even though they try their best individually and as a team. We advocate providing some hints by the White team in order to keep the learners in the exercise flow and not to get frustrated because they are stuck at one point. However, the guidance should be provided to all teams equally to preserve fair play.

- *Exercise situational awareness for learners* – the general aim of the exercise is to detect and mitigate cyber attacks. Providing exercise situational awareness for the learners might be contradictory to this aim. We provide only a basic indication of the learners' performance assessed by the White team and Red team by displaying a real-time total score of all teams on a shared scoreboard. This also proved to be an important factor fuelling participants with stress as well as a competitive mood.

- *Exercise situational awareness for organizers* – situational awareness for the White team is very important in the familiarization period where no attacks are conducted against the infrastructure defended by the Blue teams. At the beginning, all systems

are intact. Blue teams then reconfigure them to harden them and prepare the infrastructure for attacks by the Red team. The familiarization period is intentionally short so learners are under pressure and they make a number of mistakes. Monitoring the exercise's infrastructure (by the Green team) enables the White team to provide hints for Blue teams in these cases. However, this does not apply in the exercise itself because there may be states that monitoring evaluated as wrong but they were caused by a proper operational decision by a Blue team.

- *Automation of the attacks and injects* – since the exercise scenario is fixed and rigid, Red and White teams may benefit from semi-automated routines that execute the predefined attacks and injects. However, there might be an unexpected situation in which the assistance of a human operator is essential. For instance, the routines expect a file at the default location but the Blue team moved it to another place during the exercise. In addition, we are not aware of any generator of network traffic that can emulate typical Internet users, and that can be easily deployed in the exercise infrastructure.

- *Service access to the exercise's infrastructure* – to recognize an exercise infrastructure failure from scenario progression (e. g., Red team's attack or Blue team's misconfiguration), the Green team needs a service access to all sandbox components. The service access must be clearly defined in the rulebook, no attack will originate from this account, and the Red team does not have access to this account.

B.4.4 Evaluation

The exercise life cycle ends with an evaluation. It consists of an assessment of team actions and performance during the exercise, feedback survey and evaluation (after-action) workshop for the learners, and gathering lessons learned by the organizers.

The most visible part of this phase is the evaluation workshop attended by the Blue teams which lasts about a half day. Other parts of this phase are done by the White and Red teams and require much more time and preparation effort. The White team assesses e-mail communication during the exercise with respect to the non-technical learning objectives (reporting, information sharing, legal). The Red team prepares an overview of its success in attacks against particular teams and best practices related to the attacks used in the exercise. Both teams benefit from data collected by and entered into the scoring application. Furthermore, the Green team stores all collected logs during the exercise of other teams if needed. Feedback provided by the Blue teams in the survey before the evaluation workshop is also incorporated.

All parts of the evaluation (except gathering the lessons learned by the organizers) can be, again, seen as the PDCA *Plan*, *Do* and *Check* phases and the lessons as an input for the *Adjust*.

Through several runs of the exercise, we realized that learning also happens in the evaluation phase. This applies particularly to novices and learners who rated the exercise as difficult. The evaluation workshop shows the exercise scenario and timeline from the perspective of the Red team and White team. It is the only opportunity when the learners can authoritatively learn about attacks used by the Red team. They can discuss their approach in particular situations and phases. Until this point, they were only able to see the results of their experimentation during the exercise without an explanation of *why* something happened. We, therefore, recommend not to underestimate this part of the exercise and deliver analysis and lessons that will have value for the learners. For instance, a hand-out with best practices for system hardening might be useful in the daily routine of the participants.

B.4.5 Repetition

The repetition phase is an instantiation of the exercise sandbox, the execution of the existing exercise scenario for a new group of learners followed by the evaluation. Using the lessons collected in the previous phase, the repetition can be conducted with much less effort and manpower than the first run. It is also possible to skip the dry run phase after one or two repetitions. The repetition includes all phases of PDCA cycle.

B.5 Conclusions and future work

We have presented a defence exercise deployed in a cyber range and lessons learned from six runs for about 120 adult learners of various expertise, backgrounds and nationalities. The learners have no previous knowledge of the defended infrastructure and the organizers have very limited information about learner's skills and knowledge before the exercise.

We identified a general life cycle of a cyber defence exercise consisting of five phases: *preparation*, *dry run*, *execution*, *evaluation*, and *repetition*. We have described each phase and highlighted important lessons we have learned. Considering these lessons can minimize trial-and-error effort in the design, development, execution and repetition of an exercise.

B.5.1 Experience and lessons learned

Finding the best strategy to achieve a cost-effective and sustainable exercise is a very challenging goal. It is a never-ending trade-off between approaching reality and feasibility. Balancing each part of the life cycle allows the creation of a sustainable exercise that can be iteratively improved.

The preparation phase has the decisive influence on final features of the exercise. It is vital to invest many months of manpower into this phase. All systems emulated in

the exercise infrastructure must be ready including exercise content, vulnerabilities, and misconfigurations at the beginning of the exercise.

The initial version of the exercise produced in the preparation phase is not sufficient for executing successfully on its own. It must be complemented with a dry run with real learners. In our experience, the dry run verifies not only the story of the exercise but also the ability to use the exercise in repeatable deployments. Poor documentation can cause a lot of problems when making changes in a complex scenario and delay bug fixing and deployment.

Experience from the past exercises highlighted two challenges that we will investigate in our future work: *i)* how to design prerequisite testing, and *ii)* how to provide deeper feedback to the learners immediately after the exercise.

B.5.2 Future work

The limited information about prospective learners of an exercise inspired our future research on diagnostic assessment, particularly testing prerequisites for the exercise. Matching learners proficiency and exercise difficulty is a key success factor of the whole exercise. However, the best current practice is announcing the prerequisite skills and knowledge in free form, or acquiring input by self-assessment questionnaires sent out before the exercise. Both proved to be inaccurate. We are investigating methods of gaining objective information using short quizzes, tests and practical tasks related to the learning objectives of the exercise.

The scoring system produces valuable data that may be used either to compare teams mutually, or to show the progress of a team during the exercise. However, so far, the data has been aggregated to a single scoring board consisting of the current or final scores of all teams. We aim to utilize the scoring data to provide better feedback so that the learners can learn from their mistakes. We plan to present continuous scoring statistics to the learners immediately after the exercise in a well-considered interactive way and analyse their physical behaviour (e. g., eye-tracking, mouse event recording) in order to catch the interest of the learners. These techniques would expose how much feedback helps them to get insight into the passed exercise. We believe that the improved feedback from the exercise may increase learners' motivation to attend further exercises.

Acknowledgements

This research was supported by the Security Research Programme of the Czech Republic 2015-2020 (BV III/1 – VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014 – Simulation, detection, and mitigation of cyber threats endangering critical infrastructure.

Access to the CERIT-SC computing and storage facilities supported from European Regional Development Fund-Project "CERIT Scientific Cloud" (No. CZ.02.1.01/0.0/0.0/16_013/0001802), is greatly appreciated.

The Cyber Czech exercise series was designed, developed and carried out in cooperation with the National Cyber Security Centre (NCSC), a part of the National Security Authority of the Czech Republic.

Article C

KYPO Cyber Range: Design and Use Cases

Jan Vykopal[1], Radek Ošlejšek[2], Pavel Čeleda[1], Martin Vizváry[1], Daniel Tovarňák[1]

[1] Masaryk University, Institute of Computer Science, Brno, Czech Republic
[2] Masaryk University, Faculty of Informatics, Brno, Czech Republic

ICSOFT – 12th International Conference on Software Technologies. SciTePress, volume 1, 2017, p. 310-321, 12 pp.

Abstract

The physical and cyber worlds are increasingly intertwined and exposed to cyber attacks. The KYPO cyber range provides complex cyber systems and networks in a virtualized, fully controlled and monitored environment. Time-efficient and cost-effective deployment is feasible using cloud resources instead of a dedicated hardware infrastructure. This paper describes the design decisions made during it's development. We prepared a set of use cases to evaluate the proposed design decisions and to demonstrate the key features of the KYPO cyber range. It was especially cyber training sessions and exercises with hundreds of participants which provided invaluable feedback for KYPO platform development.

C.1 Introduction

Operational cyber environments are not suitable for building a systematic knowledge of new cyber threats and to train responses to them. Therefore, cyber ranges or testbeds are usually built to provide a realistic environment suitable for training security and operations teams. A cyber range provides a place to practice correct and timely responses to cyber attacks.

The learners can practice skills such as network defence, attack detection and mitigation, penetration testing, and many others in a realistic environment.

Despite the increasing popularity of cyber exercises [250, 170], there is very limited public information about platforms used. Due to the specific use of cyber ranges (government, military, industry), many technical details are regarded as sensitive. This paper shall provide an integrated view of the KYPO cyber range [135], which has been in development since 2013. KYPO was made for researching and developing new security methods, tools and for training security teams and students. It provides a virtualised environment for performing complex cyber attacks against simulated cyber environments.

Apart from the technical aspects, the transdisciplinary features of cyber exercises are equally important. Preparing and carrying out cyber exercise requires substantial time, effort and financial investments [51]. The major workload is carried out by the organizers, particularly in the exercise preparation phase. The ultimate goal of a cyber range developers is to minimize this workload and to support all phases of an exercise's life cycle. We have designed and executed a cyber defence exercise to validate the KYPO cyber range prototype. The technical part of the exercise relies on the built-in capabilities of KYPO and was used in six runs of a cyber defence exercise for 50 participants. Several lessons were learned which provided important guidance for further KYPO research and development.

This paper is divided into six sections. Section 2 shall provide background information about testbeds and cyber ranges. Section 3 will describe KYPO's architecture design and list the main components of the proposed architecture. Section 4 shall describe the user interface and interactions in the KYPO cyber range. Section 5 will show three selected use cases. Finally, Section 6 will conclude the paper and outline future work on KYPO.

C.2 Related Work

In this section, we introduce generic testbeds which can be used in cyber security. Then we focus on environments which have been specially developed for cyber security training. While some of these evolved from generic testbeds, others were designed with cyber security in mind. The environments are costly, but versatile large-scale infrastructures with state of the art parameters and features as well as lightweight alternatives with limited scope, functionality and resources.

The Australian Department of Defence published an extensive survey of state of the art cyber ranges and testbeds [60]. The survey lists more than 30 platforms which can be used for cyber security education worldwide. This number is based on publicly available, non-classified information. Since the development and operation of some cyber ranges is funded by the military and governments of various countries, there is likely to be other classified cyber ranges. To cover recent advances and innovations, we have done a systematic literature review from 2013 to 2017.

C.2.1 Generic Testbeds

Emulab/Netbed [252] – this is a cluster testbed providing basic functionality for deploying virtual appliances, configuring flexible network topologies and the emulation of various network characteristics. The network topology must be described in detail by an extension of NS language. Emulab allocates computing resources for the specified network and instantiates it in a dedicated HW infrastructure.

Emulab has been developed since 2000 and there are currently about 30 of its instances or derivates in use or under construction worldwide [76]. It can be considered to be a prototype of an emulation testbed for research into networking and distributed systems. It provides accurate repeatable results in experiments with moderate network load [216].

CyberVAN [3] – this is a cyber experimentation testbed funded by the U.S. Army Research Laboratory and developed by Vencore Labs. CyberVAN enables arbitrary applications to run on Xen-based virtual machines that can be interconnected by arbitrary networks topologies. It employs network simulators such as OPNET, QualNet, ns-2, or ns-3, so the network traffic of emulated hosts travels through the simulated network. As a result, this hybrid emulation enables the simulation of large strategic networks approximating a large ISP network.

C.2.2 Cyber Ranges

DETER/DeterLab [165] – the DETER project was started in 2004 with the goal of advancing cyber security research and education. It is based on Emulab software and has developed new capabilities, namely *i)* an integrated experiment management and control environment SEER [207] with a set of traffic generators and monitoring tools, *ii)* the ability to run a small set of risky experiments in a tightly controlled environment that maximizes research utility and minimizes risk [255], and *iii)* the ability to run large-scale experiments through a federation [81] with other testbeds that run Emulab software, and with facilities that utilize other classes of control software. Lessons learned through the first eight years of operating DETER and an outline of futher work are summarized in [23].

DETER operates DeterLab which is an open facility funded by U.S. sponsors and hosted by the University of Southern California and University of California, Berkeley. It provides hundreds of general-purpose computers and several specialized hosts (e. g., FPGA-based reconfigurable hardware elements) interconnected by a dynamically reconfigurable network. The testbed can be accessed from any machine that runs a web browser and has an SSH client. Experimental nodes are accessed through a single portal node via SSH. Under normal circumstances, no traffic is allowed to leave or enter an experiment except via this SSH tunnel.

National Cyber Range (NCR) [172] – the NCR is a military facility to emulate military

and adversary networks for the purposes of realistic cyberspace security testing, supporting training and mission rehearsal exercises [83]. Its development and operation have been funded by the U.S. Department of Defense since 2009 and the target user group are U.S. governmental organizations. The NCR enables operational networks to be represented, and interconnected with military command and control systems, with the ability to restore to a known checkpoint baseline to repeat the test with different variables. The NCR is instrumented with traffic generators and sensors collecting network traffic and data from local and distributed nodes. The NCR has demonstrated the ability to rapidly configure a variety of complex network topologies and scale up to 40,000 nodes including high-fidelity realistic representations of public Internet infrastructure.

Michigan Cyber Range (MCR) [156] – this is an unclassified private cloud operated by Merit, a non-profit organization governed by Michigan's public universities in the USA. The MCR has offered several services in cyber security education, testing and research since 2012.

The MCR Secure Sandbox simulates a real-world networked environment with virtual machines that act as web servers, mail servers, and other types of hosts. Users can add preconfigured virtual machines or build their own virtual machines. Access to the Sandbox is provided through a web browser or VMware client from any location.

Alphaville is MCR's virtual training environment specifically designed to test teams' cyber security skills. Alphaville consists of information systems and networks that are found in a typical information ecosystem. Learners can develop and exercise their skills in various hands-on formats such as defence and offense exercises.

SimSpace Cyber Range [194] – a U.S. private company runs this cyber range, which enables the realistic presentation of networks, infrastructure, tools and threats. It is offered as a service hosted in public clouds (Amazon Web Services or Google), at the SimSpace datacenter, or deployed in the customer's infrastructure and premises.

The cyber range provides several types of preconfigured networks containing from 15 to 280 hosts which emulate various environments (generic, military, financial). It is possible to generate traffic emulating enterprise users with host-based agents and run attack scenarios automatically by combining various attacker tasks. All activities can be also monitored at network and scenario level (network traffic, attackers' and defenders' actions, and activities of emulated users at end hosts). The platform is controlled via a web portal that also provides access to the results of an analysis and assessment of monitored activities within the cyber range.

EDURange [4] – this is a cloud-based framework for designing and instantiating interactive cyber security exercises funded by the U.S. National Science Foundation and developed by Evergreen State College, Olympia, Washington. EDURange is intended for teaching ethical hacking and cyber security analysis skills to undergraduate students. It is an open-source software with a web frontend based on Ruby and backend deploying virtual machines

and networks hosted at Amazon Web Services. The exercises are defined by a YAML-based *Scenario Description Language* and can be instantiated by the instructor for a selected group of students. EDURange supports Linux machines which can be accessed via SSH. It also has built-in analytics for host-based actions, namely a history of commands executed by students during the exercise.

C.2.3 Lightweight Platforms

Avatao [39, 1] – this is an e-learning platform offering IT security challenges which are created by an open community of security experts and universities. Avatao is developed by an eponymous spin-off company of CrySyS Lab at Budapest University of Technology and Economics, Hungary. It is a cloud-based platform using lightweight containers (such as Docker) instead of a full virtualization. This enables it to start a new challenge in its virtual environment very quickly in comparison with booting full-fledged emulated hosts. Learners and teachers access the challenges via web browser. Hosts and services within the virtual environment are accessed by common network tools and protocols such as Telnet or SSH.

CTF365 [56] – this is a Romanian commercial security training platform with a focus on security professionals, system administrators and web developers. It is an IaaS where users (organized in teams) can build their own hosts and mimic the real Internet. CTF365 provides a web interface for team management, instantiating virtual machines using predefined images and providing credential to access the machines using VPN and SSH. Each team has to defend and attack the virtual infrastructure at the same time. As a defender, a team has to set up a host which runs common Internet services such as mail, web, DB in 24/7 mode. As an attacker, the team has to discover their competitor's vulnerabilities and submit them to the scoring system of the CTF365 portal.

Hacking-Lab [210] – this is an online platform for security training and competitions run by a Swiss private company. It provides more than 300 security challenges and has about 40,000 users. The platform consists of a web portal and a network with vulnerable servers emulated using virtual machines or Docker containers. Each team administers a set of vulnerable applications and has to perform several tasks simultaneously, namely attack the applications of their competitors, keep their own applications secure, and up and running, find and patch vulnerabilities, keep applications up and running, and solve challenges. A Linux-based live CD is provided to ease the use of Hacking-Lab. It contains many hacking tools and is preconfigured for VPN access.

iCTF and InCTF iCTF framework [236] was developed by the University of California, Santa Barbara for hosting their iCTF, the largest capture the flag competition in the world since 2002. The goal of this open-source framework is to provide customizable competitions. The framework creates several virtual machines running vulnerable programs that are accessible over the network. The players' task is to keep these programs functional at all times and patch them so other teams cannot take advantage of the incorporated vulnerabilities.

The availability and functionality of these services is constantly tested by a scorebot. Each service contains *a flag*, a unique string that the competing teams have to steal so that they can demonstrate the successful exploitation of a service. This flag is also updated from time to time by the scorebot.

InCTF [189] is a modification of iCTF that uses Docker containers instead of virtual machines. This enhances the overall game experience and simplifies the organization of attack-defence competitions for a larger number of participants. However, it is not possible to monitor network traffic, capture exploits and reverse engineer them to identify new vulnerabilities used in the competition.

C.3 KYPO Architecture Design

The KYPO cyber range is designed as a modular distributed system. In order to achieve high flexibility, scalability, and cost-effectiveness, the KYPO platform utilizes a cloud environment. Massive virtualization allows us to repeatedly create fully operational virtualized networks with full-fledged operating systems and network devices that closely mimic real world systems. Thanks to its modular architecture, the KYPO is able to run on various cloud computing platforms, e. g., OpenNebula, or OpenStack.

A lot of development effort has been dedicated to user interactions within KYPO since it is planned to be offered as Platform as a Service. It is accessed through web browser in every phase of the life cycle of a virtualized network: from the preparation and configuration artifacts to the resulting deployment, instantiation and operation. It allows the users to stay focused on the desired task whilst not being distracted with effort related to the infrastructure, virtualization, networking, measurement and other important parts of cyber research and cyber exercise activities.

C.3.1 Platform Requirements

At the beginning of the development of the KYPO platform, many functional and non-functional requirements were defined both by the development team and the project's stakeholders. The requirements were first prioritized using the MSCW method (***M**ust have, **S**hould have, **C**ould have, and **W**ould like, but will not have*). After the prioritization process, we identified the *must have* requirements that were the most likely to influence the high-level architecture of the KYPO platform as a whole. The following selected requirements have strongly influenced our high-level design choices.

Flexibility – the platform should support the instantiation of arbitrary network topologies, ranging from single node networks to multiple connected networks. For the topology nodes, a wide range of operating systems should be supported (including arbitrary software packages). The creation and configuration of such topologies should be as dynamic as

possible.

Scalability – the platform should scale well in terms of the number of topology nodes, processing power and other available resources of the individual nodes, network size and bandwidth, the number of sandboxes (isolated virtualized computer networks), and the number of users.

Isolation vs. Interoperability – if required, different topologies and platform users should be isolated from the outside world and each other. On the other hand, integration with (or connection to) external systems should be achieved with reasonable effort.

Cost-Effectiveness – the platform should support deployment on commercial off-the-shelf hardware without the need for a dedicated data center. The operational and maintenance costs should be kept as low as possible.

Built-In Monitoring – the platform should natively provide both real-time and post-mortem access to detailed monitoring data. These data should be related to individual topologies, including flow data and captured packets from the network links, as well as node metrics and logs.

Easy Access – users with a wide range of experience should be able to use the platform. For less experienced users, web-based access to its core functions should be available, e. g., a web-based terminal. Expert users, on the other hand, should be able to interact with the platform via advanced means, e. g., using remote SSH access.

Service-Based Access – since the development effort and maintenance costs of a similar platform are non-trivial for a typical security team or a group of professionals, our goal is to provide transparent access to the platform in the form of a service.

Open Source – the platform should reuse suitable open source projects (if possible) and its release artifacts should be distributed under open source licenses.

C.3.2 High-Level Architecture

It can be seen that many of the requirements were already created with a cloud computing model in mind. This naturally influenced the KYPO platform high level architecture (Figure C.3.1). The platform is composed of five main components – *infrastructure management driver*, *sandbox management*, *sandbox data store*, *monitoring management*, and the *platform management portal* serving as the main user interaction point. These components interact together in order to build and manage *sandboxes* residing in the underlying cloud *computing infrastructure*. In the following paragraphs, we will individually describe each component. Since the user interface (platform management portal) is very complex it is thoroughly described in Section C.4.

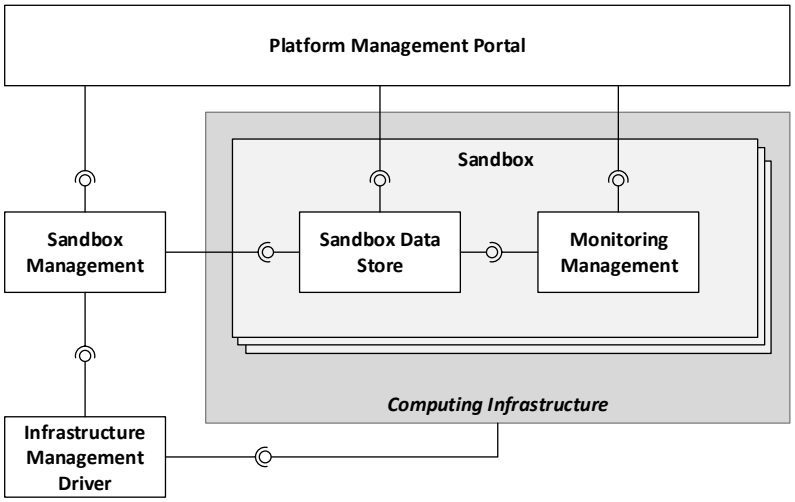

Figure C.3.1: KYPO platform high-level architecture overview.

Infrastructure Management Driver

The infrastructure management driver is used to control the computing infrastructure. A computing infrastructure consists, in general, of housing facilities, physical machines, network devices, and other hardware and related configuration artifacts. It forms the raw computing resources such as storage, operating memory, and processing power. KYPO is designed to run on public cloud computing infrastructure so that *sandboxes* can be built without the need of dedicated infrastructure.

The infrastructure management driver is the only component of the architecture which directly access the low level computing infrastructure. Therefore, the support of multiple cloud providers is isolated to this single component. API provided by the driver offers services which enable the management of virtual machines and networks in a unified way. At present, the KYPO runs on OpenNebula cloud and the adaptation to OpenStack is under development.

Sandbox Management Component

The sandbox management component is used to create and control sandboxes in the underlying computing infrastructure. During the deployment of a sandbox, it orchestrates the infrastructure via infrastructure management driver in order to configure virtual machines and networking.

Advanced networking is one of the most important features of the KYPO platform. KYPO uses cloud networking as an overlay infrastructure. The underlying cloud infrastructure uses IEEE 802.1Q, i.e. Virtual LAN tagging, using Q-in-Q tunneling. Q-in-Q tunneling

allows KYPO to configure *sandboxes* networking dynamically. It also does not depend on the L2 and L3 network addressing of the infrastructure, using a separate networking configuration. The *sandbox* networking allows users to configure their own L2/L3 addressing scheme in each LAN.

The networking in the *sandboxes* is done using one or more Lan Management Nodes (LMN). Each LAN network is managed by one LMN. LMN is a standard Debian system with an Open vSwitch (OvS) multilayer virtual switch [140]. It combines standard Linux routing and OvS packet switching. The intra-LAN communication is done on the L2 layer using OvS as a *learning switch*. The inter-LAN communication is forwarded from switch to standard Linux routing tables.

The notion of KYPO points is used to connect external devices, systems and networks to the KYPO environment. Since the KYPO platform is cloud-based, there is a need for the mechanism to be able to connect systems and devices that do not have a virtualized operating system, i. e. they are hardware-dependent, or location dependent.

We have developed a device which connects such systems – based on a Raspberry Pi platform which automatically connects after its boot via Virtual Private Network (VPN) tunnel to the sandbox in KYPO. This makes the point very easy to use since it has very small proportions and it can be easily delivered and connected anywhere. The connection is secured via the properties of the VPN.

Sandbox Data Store

The sandbox data store manages information related to the topology of a sandbox and provides its generic abstraction. Since the KYPO is partially an overlay environment, it is necessary to bridge the configuration of nodes in the cloud infrastructure and the inner configuration of virtual machines.

Therefore, modules working with sandbox-related data, e. g., the platform management portal or the monitoring management component, do not retrieve information directly from the cloud but utilize the sandbox data store instead.

The store contains information about end nodes, IP addresses, networks, routes, and network properties during the whole lifetime of the sandbox. They are updated by the sandbox management component whenever changes to the sandbox are made. For example, when a user deploys a new node or deletes a current node.

Monitoring Management Component

The monitoring management component provides fine-grained control over the configuration of the built-in monitoring and also provides an API that exposes the acquired monitoring

data to external consumers (e. g., platform management portal). All the necessary information about the sandbox's topology is read from the sandbox data store, i. e. information about existing network links and nodes. Currently, the platform supports simple network traffic metrics (e. g., packets, and error octets) and there is also support for flow-based monitoring and full-packet capture.

In order to cope with the largely heterogeneous monitoring data that is inherently generated within sandboxes and the KYPO platform itself, we use the normalizer design pattern and the notion of a monitoring bus component implementing this pattern, as described in detail by [227]. The long-term objective of such a deployment is to render the monitoring architecture within the platform fully event-driven. This is motivated by the growing need for advanced monitoring data corelations both in the terms of real-time and post-mortem analysis.

During the development of the platform, we encountered a problem as to how to differentiate between the monitoring functionality that should be built in, and the functionality that should be, conceptually, a part of a cyber exercise scenario and the resulting sandbox topology. We have determined that a reasonable decisive factor is the intended consumer of the monitoring data and the desired intrusiveness of the monitoring components on the scenario.

For example, in the case of host-based monitoring, there is a need for various monitoring agents to be installed and configured on the end-nodes. If the intended consumer is not part of the scenario, e. g., the monitoring data are used for the purposes of progress tracking or scoring in cyber-exercises, the monitoring agents must be protected from misconfiguration and other manipulation by the participants. This, however, breaks the fourth-wall, so to say, since the participants need to be informed that such misconfiguration is prohibited, including network misconfiguration and so on. This can be sometimes seen as intrusive.

When the intended consumers are the participants themselves, the monitoring components and their configuration should be a part of the scenario. This way it can be misconfigured or stopped altogether. Yet in this case, the monitoring data can be rendered unusable for external consumers, e. g., for the purposes of ex-post analysis.

Platform Management Portal

The Platform Management Portal (PM Portal) mediates access to the platform for the end users by providing them with interactive visual tools. In particular, the PM Portal is designed to cover the following types of interactive services.

Management of cyber exercises – the preparation of cyber exercises is very complex process which requires us to define security scenarios, allocate hardware resources, manage participants, and so on. The PM Portal supports the automation of these tasks by introducing a system of user roles and corresponding interactions.

Collaboration – many security scenarios are based on mutual collaboration where multiple participants share a sandbox and jointly solve required tasks or, on the contrary, compete against each other. The PM Portal supports multiple flexible collaboration modes covering a wide range of scenarios.

Access to sandboxes – the PM Portal enables end users to log into computers allocated in a sandbox via remote desktop web client as an alternative user-friendly access point to the portal-independent command line SSH access.

Interactive visualizations – regardless of whether a user is analyzing a new malware or is learning new defence techniques against attackers, it is always crucial to understand and keep track of progress and current developments inside the sandbox. The PM Portal, therefore, provides specialized visualization and interaction techniques which mediate data and events measured in sandboxes.

C.4 User Interface and Interactions

The variability of security issues that the KYPO infrastructure is able to emulate places high demands on the realization of the Platform Management Portal and its interactive services. While traditional applications are usually based on clearly defined requirements and use cases that delimit software architecture as well as provided functionality, the design of the PM Portal has to deal with the dynamic character of its usage. This is because the use cases are defined at the user level as part of security scenarios and then user interfaces have to also be either definable or at least highly configurable at the user level.

To assure high accessibility of the services for all types of end users, the PM Portal is designed as a web application where users are not bothered by the need to install anything on their device (not even browser plugins or extensions such as Java or Flash).

To deal with the dynamic character of the KYPO's use, the PM Portal complies with Java Enterprise Web Portal standards, as defined in JSR 168 and JSR 286. Web portals are designed to aggregate and personalize information through application-specific modules, so-called portlets. Portlets are unified cross-platform pluggable software components that visually appear as windows located on a web page. Once developed, a portlet can usually be reused in many security scenarios. Another key feature of enterprise web portals is their support of inter-portlet communication, synchronization and deployment into web pages and sites. We utilized these features to create complex scenario-specific user interfaces as preconfigured web pages composed of mutually cooperating portlets.

C.4.1 Role-based Access Control

Preparation of a cyber exercise is very complex task comprising scenario definition, allocation of resources, user management, and so on. In order to automatize these processes by means of user interaction, it is necessary to define user roles with clear access rules and responsibilities.

Scenarist devises security scenarios with all necessary details including sandbox definition and the design of web user interfaces for end users engaged in the scenario. At this level, the interfaces are defined as generic templates used to generate per-user web pages in further "scenario execution" phases. Besides the scenario and UI management, scenarists also authorize selected users to become organizers of exercises with adequate responsibilities.

An *organizer* is a well-instructed technically skilled person authorized by a scenarist to plan and prepare cyber exercises or experiments of a particular security scenario. Organizational activities consist of the allocation of sandboxes in the cloud, adjusting information pages, configuring a scoring subsystem and other scenario-specific services, inviting participants, etc. Organizers also delegate selected participants to be supervisors of the exercise.

Participants represent end users engaged in a particular cyber exercise or experiment. They utilize web UIs prepared by scenarists and perform tasks prescribed by the security scenario. If the users are involved in multiple experiments or exercises at the same time, they have to choose a particular one at the beginning of the interaction.

We distinguish between ordinary participants and those having extended supervising privileges. *Ordinary participants* have just one *scenario role* assigned. Scenario roles limit particular participants' access to particular hosts in the sandbox based scenario definition. For instance, an exercise scenario defines the roles of an attacker and a defender. The attacker then has no direct access to the hosts controlled by the defender and vice versa. In contrast, participants with *supervisor* privileges have access to all nodes in the network implicitly. Supervisors also usually utilize specific web forms and visualizations that reflect their specific needs. Another difference can be found in a multi-sandbox collaboration mode. While ordinary participants have access to only a single sandbox, supervisors can access all the sandboxes allocated for a given exercise.

Authentication of all users is based on federated identities. Credentials of users attempting to log into the PM Portal are redirected to a central system for identity management, which integrates many existing identity providers and authenticates users against their external electronic identities. Besides well-known identity providers (such as Facebook or Google) it is easy to integrate other external accounts on demand via a LDAP service. Participants of cyber exercises can, therefore, use their Google or corporate usernames and passwords to access the KYPO infrastructure.

Once authenticated, the authorization of a user is managed directly in the KYPO infrastructure. The PM Portal checks the user against his or her assigned roles and offers the appropriate web pages and portlets for further interaction. The more roles the user has

assigned, the broader the user interfaces of the PM Portal are available.

C.4.2 Collaboration Modes

The combination of flexible web UIs (supported by the PM Portal) and the loose coupling of individual portlets (with sandboxes via remote access) enables us to simulate various collaboration strategies [75]. Three basic collaboration modes are depicted in Figure C.4.1. The combination of these modes with other traditional web-browser features (such as multiple browser tabs opened at the same time or multi-display views) provide a very flexible solution covering a wide range of security scenarios.

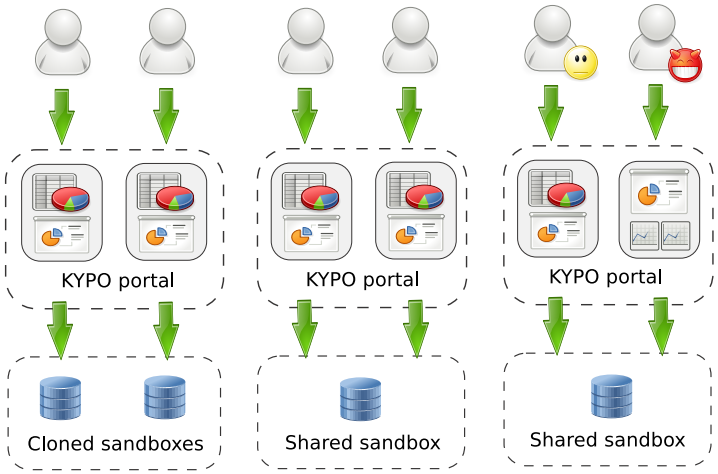

Figure C.4.1: Collaboration modes: individual sandboxes, individual views on shared data and role-based collaboration.

Individual sandboxes – every participant has their own private sandbox and web user interface. The web UI is defined by a scenarist only once in the form of a template and then the participants have the same set of interactive tools available. Thanks to its cloud-based infrastructure, it is easy to allocate many identical sandboxes for individual users on demand. Nevertheless, sandboxes do not depend on each other. Therefore, participants can complete the same tasks via the same user interface but the state of sandboxes may differ depending on their activities. This collaboration mode is useful mainly for individual training and cyber security experiments.

Individual views on shared data – the participants, each of them sitting at his or her own computer, share a sandbox and the measured data are shared. Participants have the same web UI at their disposal but they use them independently. They can focus on different parts of the network, explore different aspects of the security scenario, return back in time and so on, but they never affect the views of other participants. This collaboration mode is useful

mainly for collective learning about security threats or for collaborative forensic analyses.

Role-based collaboration – in this mixed approach, participants are divided into teams with prescribed roles, such as attackers or defenders. Teams have predefined web user interfaces according to their tasks. Teams can share a sandbox which plays the role of a battlefield. This collaboration mode is useful for exercises where multiple teams either cooperate or compete against each other in a single shared sandbox. However, this role-based approach extended with multiple sandboxes enables us to go even further. For example, we can create multiple defending teams, each having its own isolated sandbox, and a single attacking team fighting against them simultaneously.

C.4.3 Web Front-End and Visualization

The web portal technology used for the implementation of the PM Portal enables us to develop specialized interactive user interfaces, which are narrowly focused on specific goals, but also allow us to combine them easily into complex systems of mutually synchronized views supporting complex workflows. There is no space to describe all the developed user interfaces and interactive visualizations in detail, nor to discuss their combinations leading to the support of various security scenarios. Instead, we present only a few selected portlets that were used most often during various cyber exercises organized by the KYPO team so far.

Capture the Flag Games

The PM Portal offers complete support for designing and playing level-based games where users complete cyber security tasks. The administrators' interface enables the game designer to define the network topology, individual tasks, hints with penalties, time limits and other necessary information. There is also support for sandbox allocation and player enrollment. The players' interface guides them through the game and is usually supplemented with an interactive network topology view.

Network Topology

One key visualization of the PM Portal is a general topological view, as shown in Figure C.4.2. Versatility was one of the key requirements for this visualization since the network topology is present in all scenarios. Routers, links, computers and servers are represented in the visualization.

The topology visualization shows multiple dynamic data measured in the corresponding sandbox. The small icons close to the nodes represent logical roles, e. g., attacker or victim. Unusual traffic on links is visualized with colors and animations. Nodes can be accompanied by the visualization of user-defined events such as incoming emails. Clicking on a node,

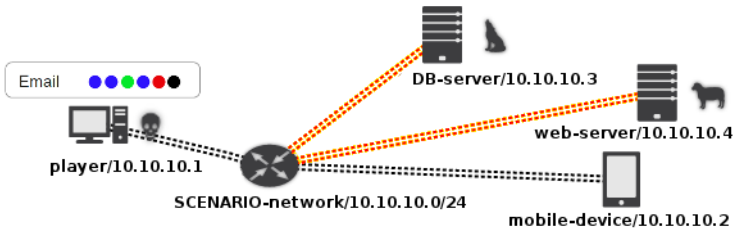

Figure C.4.2: A simple network topology with highlighted roles, network traffic and incoming emails.

a user, if privileged, can access the node's remote desktop via VNC or SPICE client. In this case, a new tab is opened in the browser with the screen of the remote host. This visualization is fully interactive, enabling users to re-organize nodes, collapse, and reveal sub-networks, zoom in and out, and so on.

Time Manager

Data measured in sandboxes and provided by the monitoring management component have the form of a time series. Therefore, many visualizations used in the PM Portal have to cope with time-related data queries in order to show the sandbox's state either at a particular point in time or within a given time span. It would be impractical to deal with time constraints in every particular portlet independently. Instead, we developed a *Time Manager* portlet (Figure C.4.3) which enable users to visually define time restrictions that are propagated to other portlets on the page.

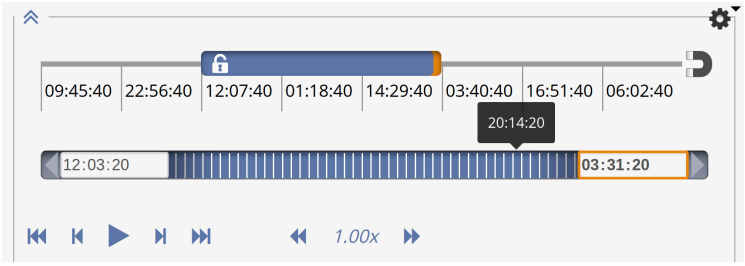

Figure C.4.3: A generic timeline management visualization.

Analytic Graphs

KYPO provides several analytically oriented visualizations and interactions. For instance, measured sandbox data can be transformed into 2D line charts and radar charts or they can be visualized in 3D, as shown in Figure C.4.4. These analytic graphs provide alternative views on multivariate data measured in the sandbox. To help users to identify anomalies,

the visualizations are fully interactive, support the re-ordering of axises simply by their direct manipulation and can switch between two 3D views smoothly by means of animation so that the user keeps track of the investigated part of the graph and never loses context.

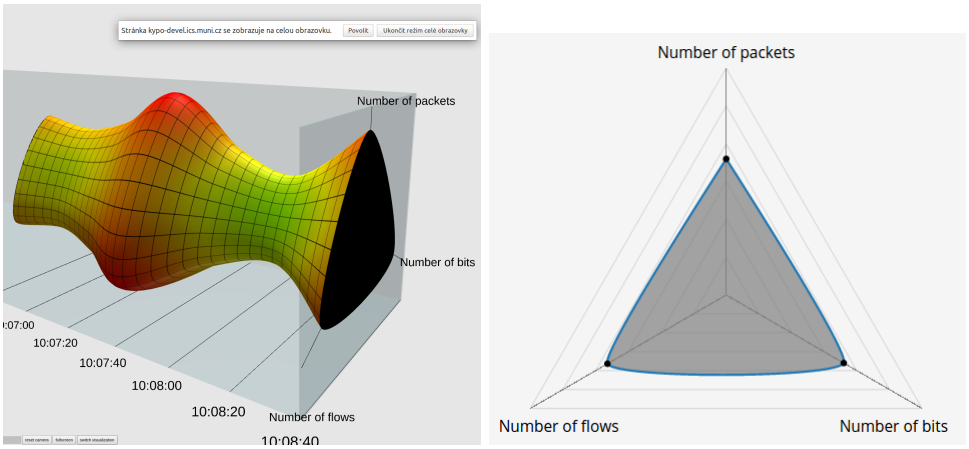

Figure C.4.4: Interconnected analytic graphs.

C.4.4 Physical Facility

Although the KYPO cyber range is accessible remotely via a web browser, many exercises are organized in a physical KYPO laboratory. Its hardware equipment offers a high variability of display techniques as well as a reconfigurability of inputs and outputs so that it is possible to support variable collaboration strategies and to distribute relevant information across several teams and roles.

The room consists of a training area over which a multimedia control center and a visitors' gallery are located, as shown in Figure C.4.5.

The training area is equipped with six mobile audio-video tables, each for 3-4 learners. The tables integrate all-in-one touch computers providing access to the KYPO infrastructure. The room is further equipped with four mobile FullHD displays and two UHD/4K displays. These displays can be either connected to individual AV tables or used to display shared information. Nevertheless, the information sharing is primarily managed by two wide central display surfaces; a projection screen and display wall. The projection screen is 5 meters wide and supports FullHD 2D or 3D projection from up to 4 sources at the same time, e. g., the supervisor's screen and working spaces of 3 teams. The display wall consists of a matrix of 5x3 Full HD displays and a multi-touch frame supporting detection of up to 10 simultaneous touches (i. e. true multi-touch).

The distribution of content into display outputs is managed centrally by the coordinator sitting in the control center. Some basic tasks can be also managed directly by the supervisor, who has a table located together with the AV tables in the training area.

Figure C.4.5: The KYPO laboratory is a versatile room. Its setting can be adjusted to best fit the needs of ongoing exercises.

C.5 Cyber Range Use Cases

KYPO can be used for various different applications. During its design and development, we focused on these three main use cases: *i)* cyber research, development, and testing, *ii)* digital forensic analysis, and *iii)* cyber security education and training. All these use cases have a similar set of requirements on the cyber range, but they differ in scenario-specific tools, the availability of pre-defined content, user interactions and expected knowledge, skills, and effort level of the users. However, the concept of sandboxes and the platform management portal helps us to cope with this fact, i. e. various types of sandboxes with various types of tools can be provided, from an empty sandbox for researchers to a fully populated and configured sandbox for a complex cyber security exercise. In the following text, we describe the differences of the three use cases in a detail, and provide references to research papers employing or benefiting from an application of the KYPO cyber range.

C.5.1 Cyber Research and Development

The first use case presented here supports research, development, and testing new methods or systems for the detection and mitigation of cyber attacks in network infrastructures of various types.

In this case, KYPO provides a sandbox and optional monitoring infrastructure for experiments. Users can provide their own virtual images for hosts to be instantiated. Alternatively, they can start with generic virtual hosts available in KYPO which run common operating systems, services and applications (e. g., Ubuntu Server, MS Windows Server 2013, Debian Server with configured DNS server) and install applications used in the experiment.

Network traffic and host based statistics can be monitored and stored within KYPO's infrastructure, where they are immediately available for analysis. Experiments can be evaluated via analytic tools that researchers deploy into the KYPO infrastructure and utilize according to their interests. Researchers can also utilize interactive visualizations of the PM Portal. The network topology visualization (with an indication of network traffic and event-based activities together with 2D and 3D analytic graphs) is especially valuable. The time manager helps to keep track of real-time developments in the sandbox.

This use case is intended for security researchers and experienced network administrators because it requires an advanced level of knowledge in networking, host configuration and some knowledge of virtualization technologies. Researchers need to be experienced in order to assemble or adjust their experiment-specific web UIs, to define their own topologies and other scenario properties, and to properly design multiple sandboxes for comparison studies. Regarding the KYPO user roles, researchers play the role of *scenarist*, *organizer* and *supervisor*.

There are several public papers using KYPO for cyber security research and development ranging from a simulation of a DDoS attack [119] through to an evaluation of a network defence strategy [158] and an analysis of surveillance software [264].

C.5.2 Digital Forensic Analysis

The second use case partially builds upon the previous one and covers basic forensic analysis, which can be partly automated by tools deployed in the sandbox. In this use case, the users can deploy virtual images of unknown or malicious machines in the predefined sandbox network and run a set of automated dynamic analyses. The sandbox contains an analytic host that provides pre-configured tools and an environment for rudimentary forensic analysis.

This use case supports security incident handlers and forensic analysts in focusing on the subject matter and removes the burden of spending their precious time in the setup of an analytic environment. Since the digital forensic analysis extends the previous use case, the required KYPO user roles are also similar.

C.5.3 Education and Training

The last use case covers a diverse type of educational hands-on activities, such as security challenges, competitions, capture the flag games, and attack/defence cyber exercises; all of

which closely follow the *learning-by-doing* principle.

In our experience, the education and training use case has proven to be the most challenging. On one hand, the KYPO platform needs to provide many additional features, mainly in the terms of user interactions, in order to support both the learners and educators in their roles. On the other hand, there is a considerable amount of customized content that must be created in order to fit a particular educational activity, whilst remaining reusable (e. g., virtual hosts, exercise data stored at hosts).

Some activities, e. g., capture the flag games, are designed to be held without much direct input from the teacher. Instead, the assignments for the learners are implanted into the platform where the game is deployed, including additional instructions and an evaluation of the submitted solutions. The learners typically choose individual tasks or follow the predefined path of the game. Once they find a solution, they submit the requested data to the game platform which immediately provides a response whether the solution is correct or not. If it is, they can proceed further.

In the other cases, it is desirable for the educators to be able to control the flow of the hands-on activity based on automatically acquired status information about the simulated infrastructure in the sandbox and also manually trigger tasks for learners, and evaluate their actions and reports.

Whatever the case, it is desirable to put minimal requirements on the learners' knowledge of the KYPO infrastructure, virtualization technologies and other advanced concepts. As a result, the learners can focus only on the subject of the exercise or training, such as a penetration testing tutorial or a cyber security game.

With regard to KYPO's user roles, learners follow the *scenario roles* assigned to them, and interact with predefined web user interfaces. Instructors have *supervisor* privileges to keep track on learners' activities and to be able to interfere in their activities if necessary. In contrast, substantial preparation effort and technical skills are required from *scenarists* and *organizers* who create the content of exercises, allocate resources and manage the preparation and execution phase.

This use case motivates further research in active learning of cyber security. We evaluated the benefits of design enhancements in generic capture the flag game scheme provided by KYPO platform [240]. Next, we introduced methods of distributing learners into teams with respect to their proficiency and the prerequisite skills required by a cyber exercise [241].

C.6 Conclusion

Today, KYPO is the largest academic cyber range in the Czech Republic. The platform is fully cloud-based and supports multiple use cases (research, education and training). We organize national cyber exercises and training sessions to validate proposed cyber range

components and to continually improve them. We also use KYPO for hands-on security courses to give students realistic experience in cyber security.

Our current work focuses on research into tools for more realistic, economical, and time efficient simulations of real cyber entities. We develop tools to further automate the preparation and execution of cyber experiments. We connect KYPO to other facilities (e. g., ICS and LTE networks) to create a more realistic cyber-physical environment. We aim to execute current and sophisticated cyber attacks in the KYPO infrastructure to provide a research environment for simulation, detection, and mitigation of cyber threats against critical infrastructure.

In addition to the technology based contributions, we would like to contribute to transdisciplinary learning in cyber security to cope with the ever-evolving threat landscape. To make a desirable improvement in the skills of the learners, technical skills must be complemented by communication, strategy and other skills for effective attack detection and response.

Acknowledgements

This research was supported by the Security Research Programme of the Czech Republic 2015-2020 (BV III/1 – VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014 – Simulation, detection, and mitigation of cyber threats endangering critical infrastructure.

Access to the CERIT-SC computing and storage facilities provided by the CERIT-SC Center, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CERIT Scientific Cloud LM2015085), is greatly appreciated.

Article D

KYPO: A Tool for Collaborative Study of Cyberattacks in Safe Cloud Environment

Zdenek Eichler[1], Radek Ošlejšek[1], Dalibor Toth[1]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic

HCI International 2015 – Human Aspects of Information Security, Privacy, and Trust. Springer LNCS, volume 9190, 2015, p. 190-199, 10 pp.

Abstract

This paper introduces the KYPO – a cloud-based virtual environment faithfully simulating real networks and enabling users to study cyber attacks as well as to train users in isolated and controlled environment. Particularly, the paper focuses on the user environment and visualizations, providing views and interactions improving the understanding of processes emerged during experiments. Web user interface of the KYPO system supports several collaboration modes enabling the participants to experiment and replay different types of security related tasks.

D.1 Introduction

Cyber attacks become more and more sophisticated and frequent. Internet users face cyber attacks on everyday basis in the form of phishing e-mails, infected attachments or intrusion attempts. A viable option to study attacks and to train users is the simulation of cyber threats in isolated, controlled, scalable and flexible cloud-based environment enabling participants to experience and replay various scenarios in order to understand the impact of the

attack on users and devices involved in the infrastructure.

There are many testbed solutions intended to support cyber security-related simulations and training programs in various manners. Some of them, namely DETER [23] and TWISC [50], employ the generic and publicly available *Emulab/Netbed* [252] infrastructure solution, which provides them with basic functionality for virtual appliances' deployment, flexible network topologies configuration, various network characteristics emulation, etc.

In contrast, several security-related testbeds require their own infrastructure solution to be established, which cannot be used for other purposes. For example, ViSe [16], LVC [231], and V-NetLab [131] testbeds employ the VMware virtualization, while the hypervisor-based security testbed [70] requires a KVM-based infrastructure. All these cases require to purchase and establish a dedicated infrastructure, which brings both strengths and weaknesses by itself – while the full control over the infrastructure can lead to easier deployment of testbed's features, it also leads to high initial costs and limited growth-flexibility. The flexibility and scalability of this lowest layer represent the key factors for possibility to create as many computer networks as needed for specific exercise scenario from the perspective of collaboration.

As another perspective can be considered integrated user environment for specific user roles and use cases. The main goal is to provide access to specific device or computer in testbed. Next important functionality is based on special visualization approaches and analytical tools, usually narrowly focused on particular aspects of network monitoring and utilized by network administrators or security analysts. The level of user interfaces (UI) differs from project to project according to its main purpose, but the majority provides only basic administration of virtual networks and users operate via traditional ways, typically SSH connections to every machine.

Next section describes the KYPO platform, which is used for management of environments for cyber security scenarios described in the paper. Third chapter briefly presents visualizations used by exercise participants for better imagination and understanding. Following chapters discuss collaboration cases of training programs, which are used in KYPO scenarios and provides user experience evaluation.

D.2 KYPO Architecture

KYPO testbed platform depicted in Figure D.2.1 provides the environment for modeling and running virtual computer networks. These networks serve as isolated environments for controlled analysis of various cyber attacks as well as for cyber security training programs [128].

Security Scenarios are employed in the whole life-cycle of cyber experiments or training programs. They represent a basic document describing the plan and necessary details

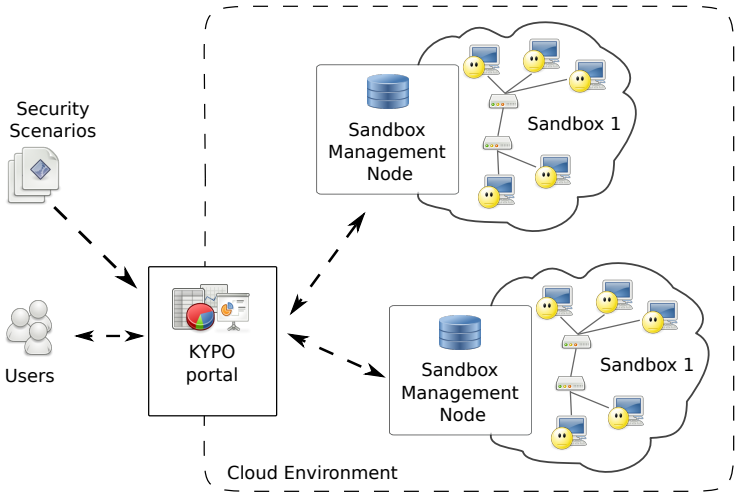

Figure D.2.1: KYPO Architecture

similarly to screenplays in movie production. Its well-structured JSON format encodes participant roles (e.g. attacker versus defender), their goals, detail instructions, roles of network nodes (e.g. mobile phone of attacker versus server to be compromised), network topology, characteristics of network links and nodes, etc. KYPO provides several predefined templates covering various security interests and domains like DDoS attack simulation, phishing, or simple hacking game. An example of a simple security scenario focused on DDoS attack simulation can be found in [119].

Network-related data encoded in a scenario are used by administrator who is responsible for the preparation of concrete training session. The scenario is uploaded to the administration interface of **KYPO portal**, which mediates access to the KYPO infrastructure for both administrators and participants. The network-related data are processed by the KYPO virtualization subsystem, which automatically allocates so called sandboxes.

Sandbox represents isolated computer network where users can safely perform their tasks. Network infrastructure of sandboxes is fully virtualized. Both nodes and links are build on top of a cloud managed by OpenNebula [163]. This approach provides scalable and flexible solution. Sandboxes can be allocated on demand and accessed remotely without the necessity to maintain hardware devices for each individual security experiment. The abstract network layers simulated by the cloud are transparent for running applications which are hardly able to detect the fact that they are not running on a physical network. The illusion of a real hard-wired network is therefore nearly perfect for both running software and users.

Once a sandbox is allocated in the cloud it can be accessed by authorized participants via KYPO portal. The portal provides users with instructions, various views on network state and also allows them to interact with the network. For example, users can connect to indi-

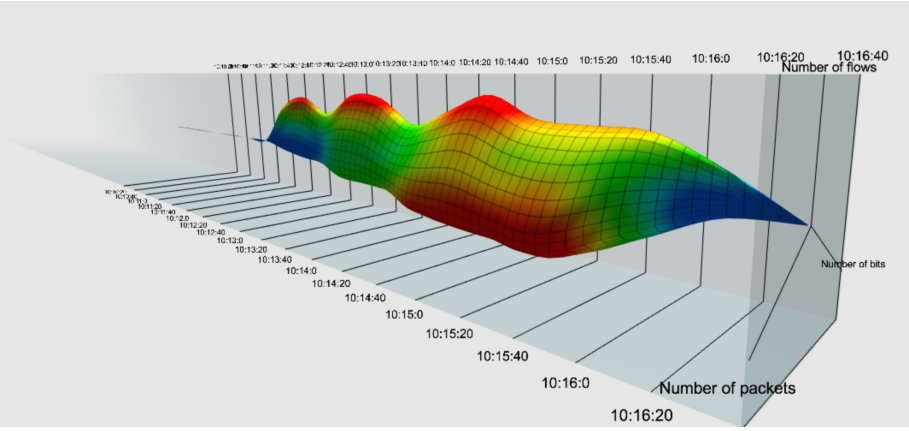

Figure D.3.1: 3D sequences radar chart

vidual computers via VNC and then launch programs and commands on them, everything via web browser.

Activities within a sandbox are monitored by probes [110, 233]. Measured data, e.g. network traffic, CPU load or security events, are stored in a database deployed in so called **Sandbox Management Node**, SMN. Every sandbox has its own SMN serving as a data repository for experiments performed in the sandbox. These data are used to provide comprehensible visual feedback to the users via interactive visualizations running at KYPO portal.

Since our tool is designed for students, sandboxes must be easy remotely accessible. Accessibility was ensured by employing the concept of Web applications with minimal requirements on web browsers. As the most fitting approach was chosen an unifying environment of enterprise portal according to its component based architecture.

D.3 Visualizations

The system provides various visualizations developed specially for educational purposes, where tutor defines which visualizations should be accessible, depending on particular scenario. All visualizations are interactive and follow the Shneiderman's visualization mantra[214]: *Overview first, zoom and filter, then details-on-demand.* One of provided visualization developed for education is a 3D sequenced radar chart, which visually compares multiple variables in time. The visualization is implemented in WebGL in order to deliver accelerated visualization in Web environment. The surface of the solid figure (Figure D.3.1) is a result of the composition of ordinary radar charts along a time scale.

A network topology visualization is presented in Figure D.3.2. Subjects of visualization

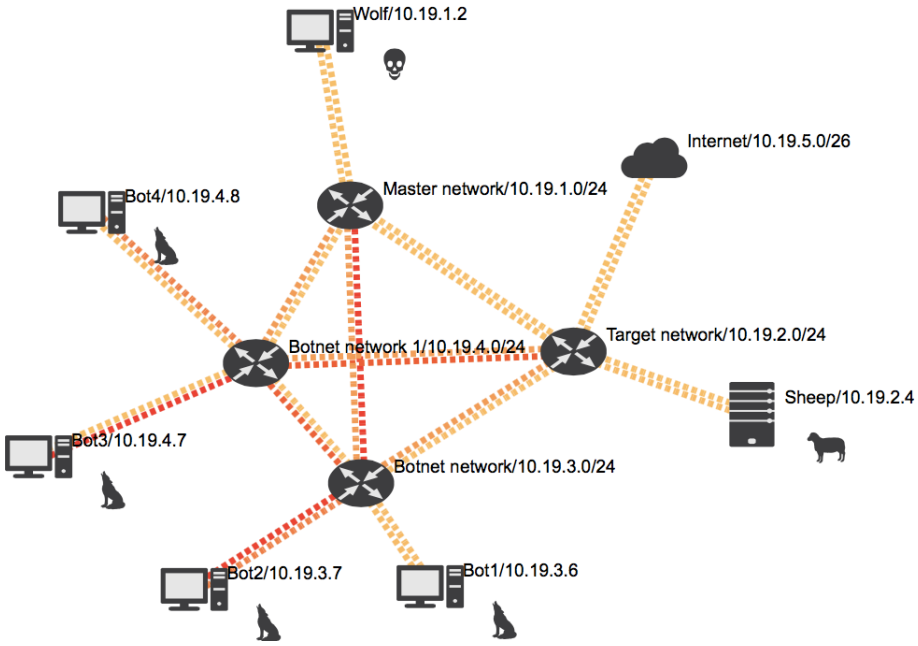

Figure D.3.2: Visualization of network topology

are routers, links, computers and servers. Every node in the topology can be accompanied by a small sign, which represents the role of the node in the running scenario (e.g. attacker or victim). Supported is also visualization of data flow on particular links. This visualization also enables students to open e.g. a VNC (remote access) connection to the computer in sandbox and share the screen of remote computer with other students or lecturer.

D.4 Collaborative Environment

Our main focus is the security training programs, where the main advantage of our approach is the authenticity. Instead of describing the key principles of cyber attacks theoretically, we rather let students to try to perform a real cyber attacks or let them e.g. to become victims of a phishing attack, all in safe virtual environment. The system provides easy to use user environment, where a lecturer is able to easily define a huge amount of attributes which will be measured in the network during cyber experiments and then presented to students either in real-time or after the experiment.

Security scenarios can significantly differ in the way how the users collaborate. There can be many sandboxes and many training programs prepared or running in KYPO at the same time. In what follows, we discuss collaboration modes of students involved in the same training session. These modes are schematically suggested in Figure D.4.1.

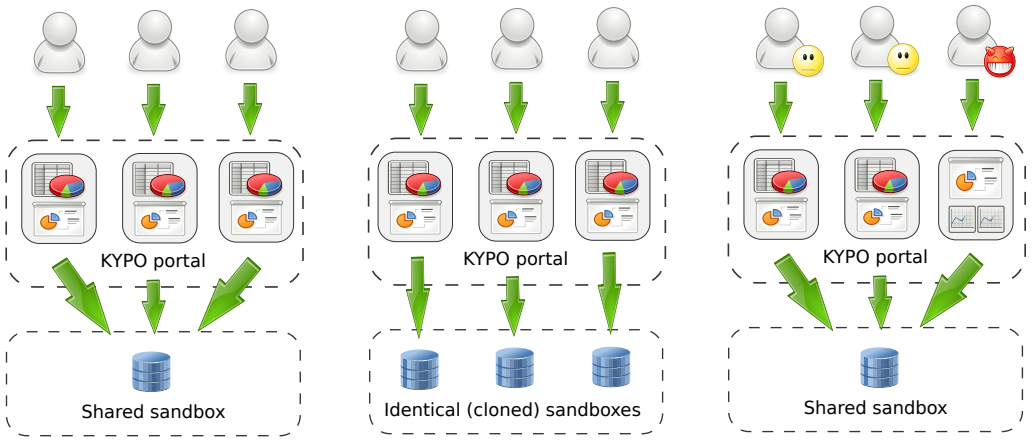

Figure D.4.1: Collaboration modes: Individual views on shared data (left), individual sandboxes (middle) and role-based collaboration (right)

D.4.1 Individual views on shared data

Imagine DDoS security scenario. It aims to illustrate principles and impacts of several variants of distributed denial-of-service attacks. The attack is driven by the lecturer who runs appropriate commands in particular sandbox. The state of the sandbox is monitored, measured and recorded in the database running on the Sandbox management node. The DDoS attack can be performed online during the training session or in advance.

Students, each of them sitting at his or her own computer, share the sandbox and the measured data. They have typically the same set of visualizations at their disposal. In the case of DDoS scenario, the most useful are the visualization of network topology emphasizing computer roles (attacker, bots, sheep) together with the utilization of links, as shown in Figure D.3.2, and also analytical tools like 3D sequence radar chart depicted in Figure D.3.1, which shows detailed link parameters on a single selected line. Although the visualizations are common for all the students involved in the training session they provide individual views on shared data. Students can focus on different links or return back in time without affecting the views of other students.

D.4.2 Individual sandboxes

Imagine a different scenario, where the users should try to compromise a computer. In this case, every participant should have its own private sandbox, in order to handle the attack on its own. Thanks to the cloud-based infrastructure of KYPO it is easy for the lecturer to allocate many identical sandboxes for individual users on demand. The sandboxes have the same network topology, network parameters, software running on nodes, and other aspects.

Events and developments of the scenario caused by users in their sandbox are measured and stored inside this sandbox. Therefore, the KYPO is able to provide per-user data after the training session.

D.4.3 Role-based collaboration

Also mixed approach is supported, where students share particular sandbox and every student has it's own role in the scenario, operating different computers. A typical example are so called "capture the flag" games, where groups of participants have access to different vulnerable computers and the goal is to compromise computers of the other groups. Another popular variant of this game defines a group of defenders protecting a vulnerable network and a group of attackers trying to compromise the network. In both the cases, students must cooperate within their groups although they are sitting at their own computers.

Security scenarios of KYPO system enables to define arbitrary roles. They also define which computer is accessible by which role. During the preparation of a training session, the lecturer of the session assigns roles to individual user accounts. The access to the computers inside the sandbox is protected by authentication. Therefore, during the session the KYPO portal provides users with the authentication data with respect to their role and the level achieved in the game.

D.4.4 Face-to-face Collaboration

Another use case, instead of the above mentioned remote collaboration, is a face-to-face collaboration. Remote collaboration through web portal enables collaboration of participants through network disregarding the geographic location. On the contrary, local face-to-face collaboration enables participants to collaborate during discussion. For this purpose, we are using the Leap Motion device, which helps participants to interactively collaborate when sharing the same computer without e.g. exchanging a mouse. The Leap Motion controller is a small USB device which captures movements of a hand performed above the device. The software recognizes particular gestures, which are then send to our visualizations. Currently, all visualizations described in Section D.3 can be controlled by the device.

D.5 Evaluation

Our system was already presented at several cyber security workshops and conferences. Online demo at NOMS 2014 conference was focused on a DDoS scenario simulated in KYPO platform [119]. More complex variant of the attack with 40 virtual machines divided into 6 sub-networks was demonstrated during the tutorial named *Cybernetic Proving Ground:*

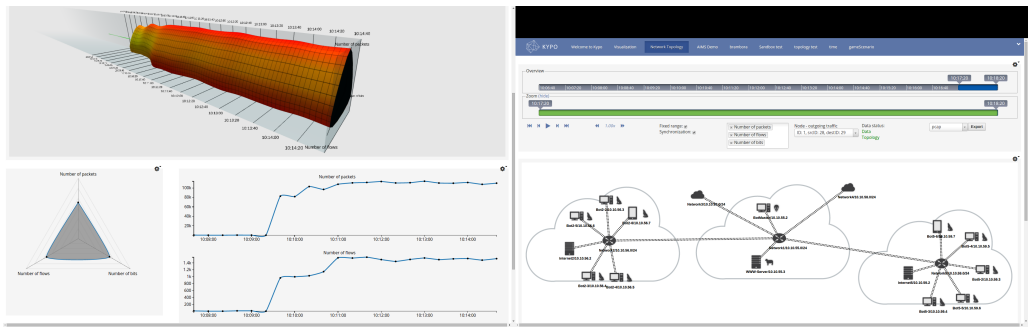

Figure D.5.1: Screen shot of the KYPO portal during the hands-on training (dual display mode)

a Cloud-based Security Research Testbed[1] attached to AIMS 2014 conference. The UI was used for the overview over the network topology traffic during the simulation and for replays of the whole scenario during the presentation for workshop members.

The second part of the AIMS tutorial was focused on a hands-on training session prepared in a form of game. The main goal was to compromise a server in a company network and to abuse it as an attacker in a DDoS attack. All 20 participants had heir own sandbox with prepared environment (several machines for this scenario) and several tasks to reach the goal (win the game). This game was successfully repeated at FIRST/TF-CSIRT Technical Colloquium[2] in 2015 with 25 involved participants.

During these workshops, no significant issues were detected. Unfortunately, no formal qualitative or quantitative feedback was collected. The formal evaluation of the KYPO system was therefore conduced on 10 university students of FI MU[3]. This preliminary evaluation brings promising results, as discussed in what follows.

D.5.1 Evaluation Process

At the beginning, subjects were asked to evaluate their knowledge about hacking (infiltration to the system) and DDoS attack. Then, subjects logged in to the system and every three or two students shared a sandbox. Instructions were provided to subjects in a form of a level based game very similar to that presented at the AIMS and TF/CSIRT Technical Colloquium, which led the students through the scenario. The goal was to compromise target system and then run DDoS attack from the compromised system. Every subject had its own computer and they were able to collaborate by sharing the screen of the attacker's computer through our web portal (VNC) and view various visualizations described in Section

[1] http://www.aims-conference.org/2014/labs.html
[2] https://www.first.org/events/colloquia/laspalmas2015
[3] http://www.fi.muni.cz

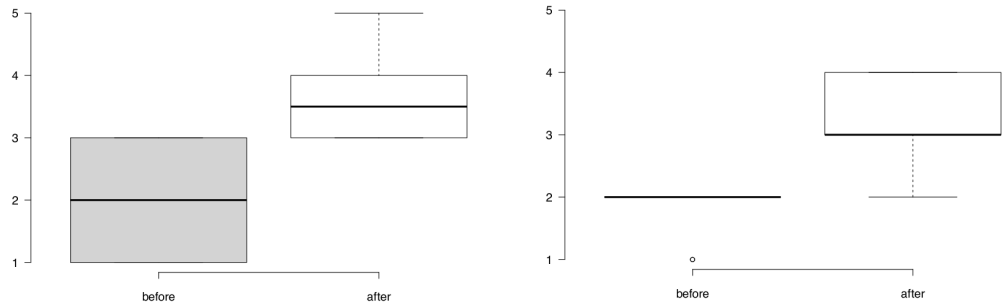

Figure D.5.2: Tukey boxplot displaying knowledge before and after the course: DDoS (left) and hacking (right)

D.3. When all subjects finished the game, the subjects were asked to the same questions again (same as before the course).

D.5.2 Results

The subjects evaluated their knowledge about DDoS and hacking on five-point Lickert scale (1 for *I don't know nothing about that*, 5 for *I'm able to perform such an attack*). The difference between before and after the course showed increased knowledge in all subjects. Comparison of DDoS knowledge and hacking knowledge is depicted in Figure D.5.2. Subjects also evaluated the course itself, also on five-point Lickert scale (1 for *Strongly Disagree*, 5 for *Strongly Agree*) on following statements (mode is a value that appears most often):

- I enjoyed the course. (mode = 4)
- I learned something new. (mode = 4)
- I enjoyed the ability to perform real attack in safe and collaborative environment. (mode = 5)

D.6 Conclusion

In this paper we have presented a cloud-based research testbed for the simulation and visualization of network attacks, focused on education and practical exercise. Chosen web-based portal technology presents a flexible and scalable solution which allows users to collaborate through various interconnected visualizations in provided web portal satisfying the requirements of broader range of training programs. Usability of our solution was verified by practical demonstrations focused on DDoS attacks and a "hacking game".

Practical evaluation and subsequent survey indicate that the proposed collaborative virtual environment equipped with user friendly interactions could be beneficial for efficient understanding of security threats as well as for the safe forensic analysis of suspicious code or devices. Our next work is therefore aimed to enhancing collaborative tactics supported by smart and intuitive interactions and visualizations.

Acknowledgments

This work has been supported by the project "Cybernetic Proving Ground" (VG20132015103) funded by the Ministry of the Interior of the Czech Republic. We appreciate the access to computing facilities *(a)* owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the program "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005), and *(b)* provided under the programme Center CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, reg. no. CZ. 1.05/3.2.00/08.0144.

Article E

Conceptual Model of Visual Analytics for Hands-on Cybersecurity Training

Radek Ošlejšek[1], Vít Rusňák[2], Karolína Burská[1], Valdemar Švábenský[1], Jan Vykopal[2], Jakub Čegan[2]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2] Masaryk University, Institute of Computer Science, Brno, Czech Republic

TVCG – IEEE Transactions on Visualization and Computer Graphics. 2021, 13 pp.

Abstract

Hands-on training is an effective way to practice theoretical cybersecurity concepts and increase participants' skills. In this paper, we discuss the application of visual analytics principles to the design, execution, and evaluation of training sessions. We propose a conceptual model employing visual analytics that supports the sensemaking activities of users involved in various phases of the training life cycle. The model emerged from our long-term experience in designing and organizing diverse hands-on cybersecurity training sessions. It provides a classification of visualizations and can be used as a framework for developing novel visualization tools supporting phases of the training life-cycle. We demonstrate the model application on examples covering two types of cybersecurity training programs.

E.1 Introduction

Our society is being exposed to an increasing number of cyber threats and attacks. The lack of a strong cybersecurity workforce presents a critical danger for companies and nations [191]. Hands-on training of new professionals is an effective way to remedy this situation. In our

work, we use visual-based sense-making and reasoning to support participants in better and faster comprehension of attacks, threats, and defense strategies.

The ability to use visual-based analytical reasoning is essential in many fields, including biology [132], medicine [137], urbanization [113], and education [93]. The goal of this paper is to create a conceptual framework providing broader insight into the application of visual analytics (VA) principles [254] in hands-on cybersecurity training. Conceptual models like the one proposed in this paper help researchers design effective visual techniques in a given domain. To the best of our knowledge, the current literature for cybersecurity training lacks such a conceptual model.

There are several reasons for the absence of a conceptual model. Existing hands-on cybersecurity training is largely heterogeneous. Training sessions differ in content, organization, target audience, and technical means. Moreover, the cybersecurity domain represents a sensitive area similar to military or intelligence services, in which many sources are secret or restricted. Therefore, it is challenging to become familiar with this domain and clarify the terms and processes. Fortunately, we have the benefit of seven years of experience with the design and organization of training sessions. The results of this paper arise from close cooperation with domain experts who directly participate in the development and operation of the *KYPO Cyber Range* [243] – a sophisticated platform for cybersecurity training. Their knowledge and the survey of other existing approaches are essential for this work.

The two most widely recognized hands-on cybersecurity training activities are *Capture the Flag* (CTF) and the *Cyber Defense Exercise* (CDX). The main difference lies in their educational goals. While CTFs focus mainly on improving hard skills in the cybersecurity domain, CDXs target both hard and soft skills. CTF features a game-like approach [236, 59, 251, 67]. Participants gain points for solving technical tasks that exercise their cybersecurity skills. Completing each task yields a text string called *flag*. In contrast, CDXs have been traditionally organized by military and governmental agencies [185] that emphasize realistic training scenarios that authentically mimic the operational environment of a real organization [73]. We deeply analyzed these types of training programs to distill a unified visual analytics model that fits the heterogeneous cyber-training events and is simultaneously instructive for the design of specialized visual analytics tools.

The major contributions of this paper are: (a) a definition of a unified training life cycle with user roles having clear responsibilities and requirements; (b) a proposal for a conceptual model of visual analytics for hands-on cybersecurity training that can be used as a framework for further research and for developing visualizations supporting particular life-cycle tasks; and (c) demonstrations of the applicability of the model using real examples and lessons learned from our long-term experience in designing and organizing hands-on cybersecurity training.

The paper is organized as follows: Section E.2 introduces the related work. In Section E.3, we discuss the generic life cycle of hands-on cybersecurity training sessions with

user roles that delimit requirements put on analytical tasks and visualizations. Sections E.4 and E.5 provide classification schemes for data and analytical visualizations. A demonstration of the conceptual model is presented in Section E.6. Section E.7 summarizes the observations attained during our research. Section E.8 outlines the direction for future research topics.

E.2 Related Work

Our work is unique in its close interconnection of three areas: visual analytics, cybersecurity, and education. Publications dealing directly with the intersection of these fields are rare. Therefore, we have explored related work from several relevant points of view.

E.2.1 Visual Analytics in Cybersecurity

Many works have addressed the challenges related to the design or evaluation of cybersecurity tools and techniques [220, 26, 17, 72, 8]. A visual analytics approach to automated planning attacks has been discussed [260]. All the surveys have confirmed the importance of supporting analytic tasks by visual interfaces. However, they are aimed at the security-related focus only and do not tackle the educational aspect of the training of new experts. We took the challenges into account in our work, and we incorporated specific aspects of hands-on cybersecurity exercises.

E.2.2 Visual Analytics in Education and Training

Another perspective that considers visualizations in relation to cybersecurity emphasizes the educational aspect. There are distinct approaches to enhancing cybersecurity abilities that focus on training or teaching computer security [208, 259, 85]. However, these works again provide outputs of a narrow scope and often omit any profound conceptualization of their findings.

To help us comprehend the topic more thoroughly, we do not focus exclusively on the cybersecurity field; we also consider studies that relate to education and training from a broader view. A recent survey [84] introduces a literature classification in the field of interactive visualization for education with a focus on evaluation, and it lists common categories of educational visualizations from distinct fields. In this respect, our work is unique as it considers more than the educational theory. It also includes the application of hands-on training with practical and technical aspects that are an essential part of the learning process.

The issue of education has been approached from the opposite direction [144]. In this work, the authors focus on predictive models for teachers of higher education institutions.

They confirm the need for insight for both the teachers and the students that exceed simple summative feedback.

E.2.3 Generic Models of Visual Analytics

Many generic design frameworks, models, and methods exist in the literature. These provide a structure and explanation of activities that designers perform when proposing suitable visualization tools [79, 154, 211, 126]. However, the aim of this paper is not to discuss processes leading to the development of specific visualizations for cybersecurity training. Instead, we provide a conceptualization of the domain so that our model can serve as a framework for discussion and the efficient application of existing design methods for specific training tasks.

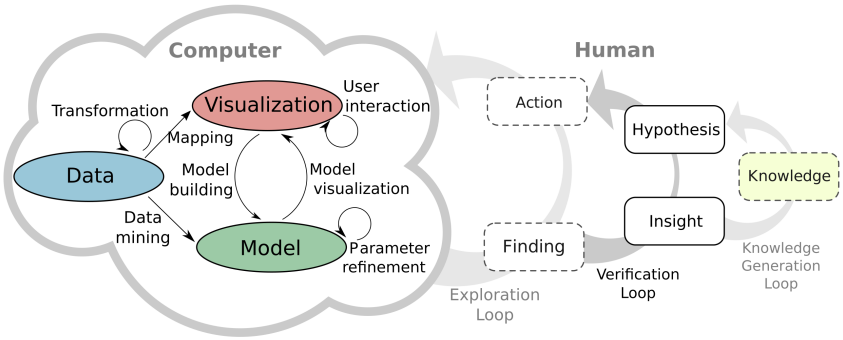

Figure E.2.1: Altered version of models by Keim [123] and Sacha [199] for insight retrieval based on visual analytics approaches.

Our solution builds upon Keim's [123] and Sacha's [199] conceptual models for the visual analytics process. The VA process is characterized by the interaction between data, visualizations, models of the data, and users discovering knowledge, as shown in Fig. E.2.1. Keim emphasizes the computer-driven components of the VA process; Sacha extends the model with human reasoning. *Data* carries facts in structured, semi-structured, or unstructured form. The *model* captures the results of automated analysis methods. The interactive *visualizations* are the primary user interface presenting *data* and *models* in a comprehensible manner. The human-centered part consists of three loops. The *exploration loop* captures low-level visual interactions using actions and findings that are specific for individual visualizations and interests. The analysts then refine their hypotheses in the *verification loop*. The *knowledge generation loop* describes the transition from observations into generalized knowledge.

These two models form the foundations of our work. We utilize *data* and *visualization* components of Keim's model and narrow our focus on the *verification loop* that plays a crucial role in building knowledge in any domain. The *model* component of the VA process

represents the cross-cutting concern, which is out of the scope of this paper. Therefore, we do not provide a separate classification for it. Instead, we mention suitable *models* in our discussion of the classification of *visualizations* and *hypotheses*. The *exploration loop* and *knowledge generation loop* are omitted since they provide either too detailed or too generic concepts.

E.3 Cybersecurity Training Life Cycle

The human loops of Sacha's VA model (see Fig. E.2.1) reflect the needs of users who interact with the computer system. Based on the literature review, our experience, and the application of analytical methods, we distilled the following general life cycle that clarifies *who* is involved in the human loops, *what* they expect (at a high level of abstraction), and *when* they conduct their VA tasks. These pieces of information are later used for the detailed conceptualization of the "computer part" of the VA model by answering *what* (data and hypotheses) and *how* (visualizations) can be analyzed in the cyber training.

E.3.1 Phases

Based on the literature review and our experience, we distilled three generic phases (see Fig. E.3.1) of the cybersecurity training life cycle. We performed a theory-driven qualitative coding method [200] on four key papers [244, 219, 125, 11] that deal with organizational aspects of cybersecurity training. Using an open coding method helped us to structure the analysis and consolidate observations. Phases and outcomes discussed in the analyzed papers can slightly differ from our model. Nevertheless, the subtleties are rather negligible since the terminology in this domain is yet not established.

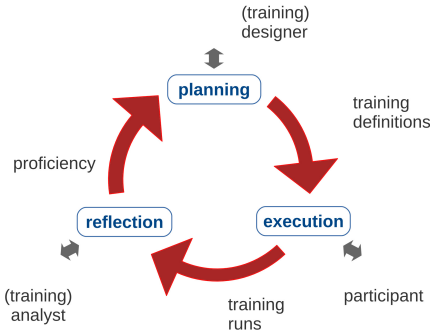

Figure E.3.1: Cybersecurity training life-cycle phases with corresponding user roles, and main outcomes of each phase.

Planning is the first phase of any new training. The goal is to formulate technical and educational requirements, set measurable objectives, and allocate necessary resources.

The *training definition* – the main output – is a set of (more or less) formally defined configurations of the computer network and its nodes, specification of attacks, training tasks and objectives, scoring rules, expected skills of participants, and related configuration data of the training.

The **execution** phase represents a training session in which participants are physically involved. User activities and the state of the training infrastructure are monitored, and the data is stored for further analysis. We refer to the data from this phase as *training runs*.

During the **reflection** phase, *training definitions* and *training runs* are analyzed and evaluated. Reflection can be conducted at any time. *Analysts* usually explore the data after each training run to learn from it or provide feedback to involved people. However, they can also analyze the data before or during the planning phase of a new training session to gradually improve its quality. The reflection phase, therefore, helps to increase the *proficiency* in designing and organizing training events.

E.3.2 User Roles

The requirements put on visual analytic interfaces are affected by user roles. The basic roles emerged from the life cycle. They reflect individual phases captured in Fig. E.3.1. For clarity, our roles are CAPITALIZED in the paper.

Training designers (**designers** for short) are responsible for the design of training definitions during the *planning* phase. Multiple designers with different skills are usually involved in the preparation of new training content. Cybersecurity experts contribute primarily to the technical aspects; education experts are responsible for defining the learning objectives and assessment criteria.

Participants represent everyone involved in the training event. Their analytical activities are associated with situational awareness and gaining insight into the training during the *execution* phase.

The **training analyst** (**analyst** for short) role covers all the people who conduct the post-training analysis of collected data. In our VA model, this role is used to capture the requirements of generic analytical interactions. Various people interested in the relevant data can take on this role, e.g., cybersecurity experts looking for talented participants.

These three roles are not independent. Arrows in Fig. E.3.2 represent the inheritance of user roles as defined by requirements analysis methodologies in software engineering [206]. It means that DESIGNERS and PARTICIPANTS can conduct post-training analysis like other TRAINING ANALYSTS, e.g., to get feedback on completed training sessions. On the other hand, they can have a specific responsibility during the *planning* or *execution* phases, respectively.

The high-level roles that emerged from the life cycle proved to be too general to capture

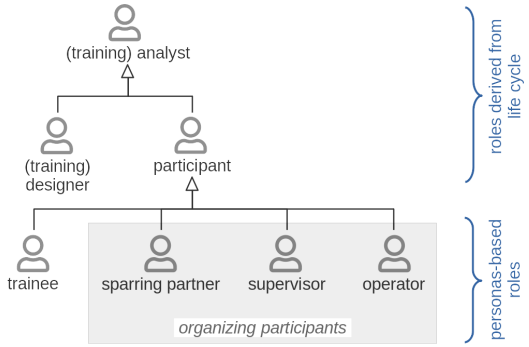

Figure E.3.2: Hierarchy of user roles participating in cybersecurity training.

the fine-grained requirements of heterogeneous groups of people participating in real training events. Therefore, we employed the personas design method [102] to reveal archetypal users and further decompose user roles. We analyzed the same sources that we used during the conceptualization of the life cycle [244, 219, 125, 11]. The observed personas are summarized in Table E.3.1.

CTF training includes only two types of personas, which correspond to a teacher-student relation. The *student* (or *learner*) follows instructions defined by the *training definition* and performs the required tasks. The *instructor* facilitates the training session from the educational point of view. Moreover, the instructor is also responsible for the technical aspects of training and addresses any possible technical difficulties with the underlying infrastructure.

In CDXs, we identified seven personas. *Blue team* members are similar to *learners* of CTFs. They have to defend the entrusted network from the attacks of the *red team*. *White team* members are responsible for the organization and compliance with the "game rules" of a CDX. *Fictitious users* represent common users of the defended network. *Law enforcement officers* check whether the actions of the *blue team* are legal. *Journalists* request reports from the *blue teams*. Finally, the *green team* is responsible for maintaining the infrastructure of the exercise.

By deeply analyzing the responsibilities and analytical goals of identified personas, we generalized them to four user roles. The mapping is captured in Table E.3.1.

Trainees solve tasks described in the *training definition*. Their activities are monitored and assessed. They can work either individually or in teams. For the sake of simplicity, we use the term "trainee" for both cases.

Sparring partners represent individuals or teams involved in training sessions who actively compete with TRAINEES but who are not directly assessed. Sparring partners also follow the instructions from the *training definition*. However, their requirements for data analysis, feedback, and other educational aspects differ from the requirements for TRAINEES.

Table E.3.1: Mapping of CTF/CDX personas to fine-grained user roles.

user roles	CTF personas	CDX personas
trainee	student (learner)	blue team
sparring partner	–	red team
		white team
		fictitious user
		law enforcement officer
		journalist
supervisor	instructor	green team
		white team
operator	instructor	green team

Supervisors, unlike SPARRING PARTNERS, do not follow the exact rules of the *training definition*. They are responsible for overseeing the training session, enforcing rules, and other activities that are not exactly defined.

Operators are responsible for the underlying (technical) infrastructure of the hands-on training. This role requires technical skills and a good knowledge of the underlying technologies. The work of operators can significantly affect the course of the exercise since any technical difficulties can devalue educational results regardless of how well the training session has been prepared.

All the roles distilled from personas represent participants directly involved in a specific training session. Therefore, they are defined as descendants of the PARTICIPANT role in the schema in Fig. E.3.2. While TRAINEES are the primary subject of training sessions, SPARRING PARTNERS, together with SUPERVISORS and OPERATORS, represent backstage *organizing participants*.

E.4 Data

Visualizations designed for operational cybersecurity deal with large data sets [26]. In contrast, training events are limited in time, resources, and the number of participants. As a result, the amount of data produced during the training sessions is also usually limited. However, the data is highly heterogeneous. Therefore, our classification has been developed iteratively together with the analysis of other parts of the VA model. The proposed scheme comes from the unified life cycle. Data categories reflect user roles and training phases during which the data is created. It enables us to clarify what data is available in each phase and define limitations to be considered in analytical visualizations.

Technical scenarios ($\mathbf{D_1}$) capture the technical aspects and predefined processes of a *training definition*. The technical aspects include, for example, the definition of the network topology, software running on individual network nodes (operating system, applications, services), and vulnerabilities injected in the network nodes. User procedures are defined as

attack plans (attack vectors and their timing), trainees' tasks, hints, and other formalized steps.

Assessment criteria (D_2) determine how to assess trainees and how to measure whether learning objectives were achieved. Assessment criteria define metrics, indicators, and aspects of the training related to the evaluation of trainees. Apart from that, the criteria can also include the definition of questionnaires for prerequisite testing of trainees, assessment questions during the exercise, and post-training feedback surveys.

User actions (D_3) are participants' actions monitored and collected during the *execution* phase. Examples include commands entered by trainees, displayed hints, performed attacks or defenses and their results, intervention of supervisors, and other user-oriented events.

Infrastructure data (D_4) represent the state of computer networks and the underlying technical infrastructure. The data encodes node availability, available services, packet flows, and the health of the infrastructure. The obtained information can be used for direct infrastructure surveillance, and the assessment of trainees (e.g., trainees can be penalized for the unavailability of required services).

Assessment data (D_5) are related to the *assessment criteria* and determine the success rate of trainees and their results in achieving learning objectives. The data encodes how successfully a particular user has solved a particular task (in percentages or as obtained penalties), time spent on tasks, answers to questionnaires, and other qualitative and quantitative indicators of the learning process. A great deal of quantitative data can be computed automatically by applying assessment criteria (D_2) to monitored user actions and infrastructure data (D_3 and D_4).

Table E.4.1: Data types mapping on life cycle phases, abstract data levels, and terminology from the paper.

	D_1 & D_2	**D_3 & D_4 & D_5**
phase of creation	planning	execution
level of abstraction	configuration data	operational data
terminology	training definition	training run

Mapping data categories to the planning and execution phases follows data abstraction as defined by Fowler for software systems [86]: D_1 and D_2 represent data from the *configuration level*. They are defined during the *planning* phase by designers as a part of *training definitions*. D_3–D_5 represent data from the *operational level*. They are acquired during the *execution* phase and we refer to them as *training runs*, as summarized in Table E.4.1.

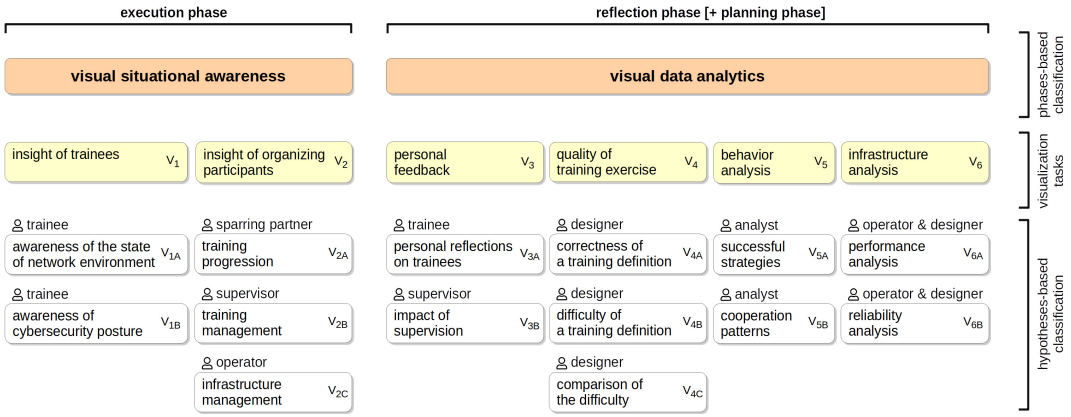

Figure E.5.1: Classification of visualizations and hypotheses in the context of hands-on cybersecurity training.

E.5 Visualizations and Hypotheses

According to the VA model of Sacha & Keim (see Fig. E.2.1), requirements applied to visualizations are driven by hypotheses that people consider during their analytical activities. Therefore, we discuss and classify both visualizations and hypotheses together.

The classification shown in Fig. E.5.1 was established iteratively by balancing two complementary directions. We broke down the top-level phases and roles of the training life cycle and, concurrently, we searched for low-level hypotheses that we organized into clusters. Balancing these two approaches, we concluded with a three-level classification scheme that, to the best of our knowledge, sufficiently covers the problem domain and emphasizes the design requirements of visual analytic tools. The low-level hypotheses were obtained from discussions with six domain experts (three of them are co-authors of this paper), each with more than six years of experience with organizing CTFs and CDXs. The final classification hierarchy was reached by consensus of the authors whose expertise includes cyber training design and organization as well as the design of analytical visualizations for KYPO Cyber Range [243]. The rest of this section is structured according to the proposed scheme as follows.

The top-level categories of *Visual Situational Awareness* and *Visual Data Analytics* in Fig. E.5.1 represent distinct concepts using different data in different phases of the life cycle. They are discussed in two separate subsections. During conceptualization, we observed that the analytical tasks of TRAINING DESIGNERS represent a subset of activities associated with the *reflection* phase of TRAINING ANALYSTS. Hypotheses and visualizations of the *planning* phase are, therefore, covered by the *Visual Data Analytics* category.

Classification at the second level defines key visualization tasks V_1–V_6 that are detailed later in this section. They differ in the roles involved in the visual analysis, analytical goals,

and other aspects. Discussion is primarily focused on visual requirements and justification for the third-level classification of hypotheses V_{1A}–V_{6B}.

Providing an exhaustive list of hypotheses for each task V_{1A}–V_{6B} is impossible; they emerge continuously as users conduct analyses and gain insights into the solved problem. Instead, we discuss an abstraction used for the classification and propose several hypotheses as examples.

E.5.1 Visual Situational Awareness

Existing theoretical concepts of situational awareness distinguish between *perception*, *comprehension*, and *projection* corresponding to the three levels of the well-known Endsley model [78]. However, the significance and meaning of the levels can differ in the context of cybersecurity training depending on users' roles and their goals. This is because providing comprehensive insight into cybersecurity events during the *execution* phase can be undesirable in certain circumstances. This aspect is reflected in our classification, as discussed in what follows. Table E.5.1 summarizes visualizations and hypotheses for situational awareness.

Insight of Trainees (V_1) visualizations support TRAINEES in keeping track of what is happening at the moment and understanding the training content. The view on the data should be strictly person-centered and adapted to the history and performance of each particular TRAINEE so that they can concentrate on the development during the training session from their perspective.

The level of detail provided to TRAINEES has to be carefully considered when designing visualizations. A visual storytelling approach to learning can provide comprehensive guidance of TRAINEES throughout the training session. Using event-based visualizations emphasizing important actions and events that appeared during the *execution* phase can help the TRAINEES grasp the main ideas of the training content. However, this approach is rather exceptional, and visual guidance is usually intentionally restricted. A typical goal of hands-on cybersecurity training is just to exercise the *perception*, *comprehension*, and *projection* skills of TRAINEES; a subtle visual run-time support better mimics real-world conditions. The visual-based comprehension is often left for the *personal feedback (V_4)* tools in the *reflection* phase (discussed later in this Section).

The clustering of hypotheses revealed two fields of TRAINEE interest. *Awareness of the state of the network environment (V_{1A})* covers hypotheses relevant to overseeing the state of the training network maintained by a TRAINEE. It is used to infer knowledge of hidden cyber events and actions from the *infrastructure data (D_4)*. *Awareness of cybersecurity posture (V_{1B})* is related to the understanding of cyber events and actions defined as education goals in *training definitions*.

Insight of Organizing Participants (V_2) visualizations support SPARRING PART-

Table E.5.1: Visual Situational Awareness: Visualization tasks V_1 and V_2 are further divided into two (V_{1A}–V_{1B}), and three (V_{2A}–V_{2C}) categories. Each category is accompanied by sample hypotheses formulated as prerequisites for verification ("I suppose that ...").

V_1 – Insight of Trainees
Awareness of the state of network environment (V_{1A}):
As a *trainee*, I suppose that ...
... the web running at host X is accessible for users.
... the host X is accessible for me via SSH.
... the external network (including internet) remains accessible.
Awareness of cybersecurity posture (V_{1B}):
As a *trainee*, I suppose that ...
... server X I am defending is now under attack.
... my previous attack actions were successful.
... I have successfully protected server X against the DDoS attack.
V_2 – Insight of Organizing Participants
Training progression (V_{2A}):
As a *sparring partner*, I suppose that ...
... the trainee X completed task Y, a prerequisite for task Z.
... the DDoS attack against host X defended by trainee Y was successful.
... trainee X fixed the vulnerability allowing a DDoS attack at host Y.
Training management (V_{2B}):
As a *supervisor*, I suppose that ...
... all trainees completed task Y, a prerequisite for task Z.
... trainee X solved the task successfully.
... trainee X is in trouble (working on task longer than Y min).
Infrastructure management (V_{2C}):
As an *operator*, I suppose that ...
... service X at host Y is up and running.
... service X at host Y is inaccessible longer than Y min.
... network of trainee X is connected to the rest of exercise infrastructure.

NERS, SUPERVISORS, and OPERATORS in gaining insight into the state and progress of training sessions. Views are usually shared across all participants of the same role, providing them a view of the training progression, score, solved tasks, and other milestones and assessment data related to planning and timing. However, the views have to be adapted to each organizing role. V_2 is, therefore, divided into three categories of hypotheses according to organizing roles. *Training progression (V_{2A})* is used by SPARRING PARTNERS who need to know the current state of the TRAINEES' networks and services so that they can coordinate their actions and perform them in proper order and time. *Training management (V_{2B})* of SUPERVISORS should be able to identify troubles of TRAINEES as soon as possible. *Infrastructure management (V_{2C})* is intended for OPERATORS who have to monitor the unreliable infrastructure of the cyber range to detect technical problems.

Regardless of the specific role, the supervising activities of all organizing participants force them to *perceive* the current state of the training, to *comprehend* the situation, and to *project* the future status so that the training progresses smoothly and efficiently. In contrast to the *Insight of Trainees (V_1)*, analytical visualizations of organizing participants should fully support all these levels of awareness.

E.5.2 Visual Data Analytics

Our classification combines user roles of the cybersecurity training life cycle (see Fig. E.3.1) and data categories (Section E.4). Table E.5.2 summarizes the classification of hypotheses that are explained in the remainder of this section.

Personal Feedback (V_3) to participants has a significant positive impact on the learning process [184, p. 480]. A good post-training visual feedback should explain the pros and cons of the chosen approach and indicate the areas for further improvement.

Effective person-centered feedback should occur as soon as possible, during or right after the *execution* phase when the trainees remember details of their behavior, decisions, and conducted actions. Deploying such immediate visual feedback requires automated data processing and automatically generated personalized views for individual trainees.

Our classification scheme is divided according to roles that benefit from timely feedback: *personal reflection of trainees (V_{3A})* and *impact of supervision (V_{3B})*.

Personal feedback is crucial for the trainees to learn from the exercise as much as possible. Nowadays, the feedback is often restricted to providing a simple scoreboard with very limited informal comments from supervisors (a so-called "hot wash-up" session). There might be an additional debriefing later when supervisors manually process the data. However, the analysis is laborious, and the delayed presentation of findings might reduce the impact on trainees [244]. They should receive a view of their behavior during the training session as well as comparison with other trainees. Moreover, the data analysis should be automated to provide in-depth feedback right after the training session. Feedback visualizations have to be well-designed and intuitive. Using common techniques would be necessary because trainees usually do not have time to familiarize themselves with complex tools. A low number of easy-to-decode charts (bar/line charts, scatter plots, etc.) should be favored over the complex VA tools. The user interface should motivate users to explore the data and learn from their mistakes. Applying the methods of user-centered design [174, 154] is, hence, a must.

Supervisors can also benefit from personalized feedback after a training session since their interventions influence trainees. The visualizations should provide an overview as well as detailed per-trainee data. This allows supervisors to analyze the impact of their interventions and learn from their possible mistakes in managing the training session.

Table E.5.2: Visual Data Analytics: Visualization tasks V_3–V_6 are further divided into several categories (e.g., V_{4A}–V_{4C}).

V_3 – Personal Feedback
Personal reflection of trainees (V_{3A}):
As a *trainee*, I wonder ...
... what I did wrong in the task X.
... where I lost the most points and why.
Impact of supervision (V_{3B}):
As a *supervisor*, I wonder ...
... if I intervened in time.
... if I intervened properly.
V_4 – Quality of Training Exercise
Correctness of a training definition (V_{4A}):
As a *designer*, I suppose that ...
... all tasks are relevant to learning objectives.
... task X of the training definition Y is solvable.
Difficulty of a training definition (V_{4B}):
As a *designer*, I suppose that ...
... prerequisite skills of trainees were well-defined.
... the training definition X is suitable for beginners/experts/...
Comparison of the difficulty (V_{4C}):
As a *designer*, I suppose that ...
... the training definition X is more difficult than definition Y.
... tasks in the training definition X require more time to finish than tasks in definition Y.
V_5 – Behavior Analysis
Successful strategies (V_{5A}):
As an *analyst*, I suppose that ...
... limiting network access is a better strategy than fixing individual vulnerabilities in the network.
Cooperation patterns (V_{5B}):
As an *analyst*, I suppose that ...
... closer cooperation between team members leads to more effective protection against attacks.
... the team X had a strong leader who communicated with the rest of the team significantly more often.
V_6 – Infrastructure Analysis
Performance analysis (V_{6A}):
As an *operator* or *designer*, I search for ...
... the most utilized links/nodes/CPUs in the infrastructure for training definition X.
... the peak memory usage of individual network nodes in training definition X.
Reliability analysis (V_{6B}):
As an *operator* or *designer*, I search for ...
... the mean time to failure of nodes in the infrastructure.
... unstable custom network services in the infrastructure.

Feedback for SPARRING PARTNERS and OPERATORS is rare, since the main objective

of the training is to teach TRAINEES. This is why we omitted these two roles from the classification.

Quality of Training Exercise (V_4) reflects the usefulness of training sessions for TRAINEES. The main motivation is to improve future training programs by reviewing collected data by DESIGNERS, i.e., experts with educational skills, who are responsible for the training content. The quality can be measured and compared by various qualitative attributes that capture individual features of training sessions. *Correctness*, for example, can express the ability of TRAINEES to solve required tasks considering properties of the underlying infrastructure, the logical consistency of tasks, or availability of meaningful instructions. *Difficulty* can be expressed as the time required to finish the training session or minimal skills required of TRAINEES. DESIGNERS can study either results of individual *training runs* of the same *training definition* or compare *training definitions* mutually.

Our classification scheme divides V_4 hypotheses according to qualitative attributes and the multiplicity of involved training runs: *Correctness of a training definition (V_{4A})*, *difficulty of a training definition (V_{4B})*, and *comparison of the difficulty (V_{4C})*. Other qualitative attributes, apart from correctness or difficulty, can be considered. However, not all combinations are meaningful. For example, correctness typically represents a binary value (correct or incorrect) and then mutual comparison does not make sense.

The quality of a training session is primarily affected by three mutually connected factors:

- Training content defined by *technical scenario (D_1)*. Ambiguous or illogical tasks and their extreme difficulty or simplicity can discourage TRAINEES from proceeding, rendering the training session useless.
- Assessment defined by *assessment criteria (D_2)*. They affect achieving educational goals. Unbalanced assessment (too lax or strict) can lead to bypassing tasks or demotivate TRAINEES.
- Proficiency and motivation of TRAINEES. The lack of knowledge, skills, or motivation can prevent TRAINEES from finishing the training. Knowledge and skills are usually measured as part of prerequisite testing using questionnaires or small practical tasks.

Visual analytics can help to balance these factors by providing different views on the triplet and enabling DESIGNERS to study their mutual interactions and dependencies so that the impact of training is maximized for a given group of TRAINEES. Techniques of multiple coordinated views [192] can be used to support this exploratory analysis effectively.

Behavior Analysis (V_5) can help in discovering relevant facts about TRAINEES, their skills, or behavioral patterns under stress. The observations can either reveal issues or inconsistencies in training definitions or identify general patterns applicable in practical cyber defense. For instance, visualization of users' actions can reveal patterns of successful cooperation or successful attack/defense strategies.

Successful strategies (V_{5A}) and *cooperation patterns (V_{5B})* are two primary categories of analytical hypotheses directly related to cybersecurity education where visual perception can significantly help. The former analyzes defense and attack strategies, e.g., completely cutting off the defended network on the firewall vs. selective suspension of services being under attack. The analysis of cooperation patterns can be considered a part of the strategy analysis. However, it focuses more on people, their cooperation tactics, and how they influence the results of the training. The classification scheme can be extended to reflect other requirements of cybersecurity experts.

The raw data D_3 – D_5 of *training runs* has usually a form of time-stamped events. Reconstruction, visualization, and analysis of user processes that produced the data are possible by employ techniques of process mining [248, 130]. Analysis of behavioral aspects can also be supported by specific statistical, knowledge discovery, or machine learning *models* incorporated into the VA process (see Fig. E.2.1). For example, methods related to the node centrality in social networks [176] can be used to identify skilled leaders in team-based training sessions. Anomaly detection algorithms [46] can identify strong/weak skills of *trainees*, for instance.

These data can also serve to measure learning. [150] proposes several metrics for measuring performance that are applicable in cybersecurity training. These include tracking the time spent on tasks, observing the usage of specific tools in logs, or automatically checking properties of the virtual environment, such as uptime of services. A concrete example in the context of CDXs is presented in [147]: the evaluators measure the time of the attack, compromise, detection, mitigation, and restoration. In [108], also non-technical aspects are measured, such as team behavior.

Infrastructure Analysis ($\mathbf{V_6}$) represents another essential activity that can affect the results and impact of cybersecurity training. Any technical difficulties or malfunctions can negatively influence TRAINEES. Related visualizations should support OPERATORS and DESIGNERS in exploring *training definitions* and their requirements on the infrastructure and provide them with a "backstage" view on the operational data captured in the *execution* phase.

As opposed to the *infrastructure management (V_{2C})* in situational awareness, this category relates to the feasibility of the underlying infrastructure to serve according to the prescription of the *training definitions*. For example, if a heavily used server is allocated on a shared virtual node in the cyber range, then its response time can be prohibitively slow. This can hinder TRAINEES in fulfilling the tasks.

Suitable visual tactics strongly depend on features and possibilities that are specific for technology used to implement the underlying infrastructure. Our classification, therefore, uses qualitative aspects that delimit generic requirements on the infrastructure: *performance analysis (V_{6A})* and *reliability analysis (V_{6B})*. The performance deals with the utilization of resources at various levels of granularity (CPU, memory, network nodes). Reliability is

related to the failure rate of individual facilities. However, these two qualities represent only an example.

E.6 Demonstration

In this section, we illustrate the application of our conceptual model on the KYPO Cyber Range platform, which is being developed by the cybersecurity team at our university since 2013. From the beginning, KYPO was designed with an emphasis on user-friendliness and support for providing interactive visual insight into cybersecurity and learning processes. It represents a comprehensive system suitable for demonstrating the applicability of our model. As the KYPO visualizations were designed on the fly without a conceptual view towards the application domain, this section aims to demonstrate how the model fits the existing design of a complex cyber range and to reveal the undersupported parts of the training life cycle. The presented visualizations only illustrate possible approaches to the design of specific visual analysis tools.

To the best of our knowledge, other cyber ranges and cybersecurity training tools focus primarily on the training content, providing only limited visual insight. Nevertheless, we aim to discuss other approaches when the KYPO does not provide a suitable example.

E.6.1 Training Life Cycles and Data in KYPO

The **KYPO Cyber Range** [243] is a highly flexible and scalable cloud-based platform. Its core functionality is to emulate computer networks with full-fledged operating systems and network devices that mimic real-world systems. Its primary use is hands-on cybersecurity training, especially *attack-only* capture the flag games and cyber defense exercises. It is also used in other cybersecurity applications, such as forensic investigation. The platform provides tools for the automated collection of various data that can be further analyzed. These include network flows, computer logs, user commands, and user actions from GUI (e.g., mouse clicks or submitted forms).

The main user interface is a web application called the *KYPO portal*. We gradually extend the set of available visualizations and visual analytics tools integrated into the *KYPO portal* using the participatory design process. Nine cybersecurity experts (two specializing in cybersecurity education who are co-authors of this paper) closely collaborated in the design and evaluation of novel visualizations and the improvement of their features.

Capture the Flag games consist of tasks divided into consecutive levels where access to the next level is conditioned by completing the previous one. Players can use hints or skip entire levels. These actions (taking hints and skipping or completing a level) are penalized or rewarded by scoring points. The final scores of individual trainees within the same session are mutually comparable and can be used for their evaluation. A typical session

lasts for one to two hours. Several SUPERVISORS facilitate a group of up to 20 TRAINEES working as individuals or in pairs.

DESIGNERS of CTF games are experts from the cybersecurity incident response team of our university or undergraduate students of a one-semester course on designing cybersecurity games [239]. They produce *training definitions* that describe both *technical scenarios (D_1)* and *assessment criteria (D_2)*. The training definition is a set of (plain text) documents that include: a description of the network environment and the configuration of individual network nodes (including vulnerabilities to be exploited in the game levels); a common background story and task descriptions (for each level); definition of hints, worked-out solutions and penalty points for taking hints (for each level); the TRAINEE's prerequisites, educational objectives and further assessment criteria. *Designers* can interactively prepare content and allocate resources required for training sessions through the KYPO portal.

The produced *training definitions* are used for creating training sessions in the *execution* phase. The KYPO Cyber Range automatically logs TRAINEES' *user actions (D_3)*. Some of the *training definitions* contain pre- and post-game questionnaires for assessing TRAINEE knowledge (i.e., *assessment data (D_5)*), which is stored as well. So far, *infrastructure data (D_4)* collection is not supported in CTF games.

Cyber Czech is a series of technical cyber defense exercises for up to six *blue teams* (3–4 members). The TRAINEES must protect their infrastructure against various attacks from the *red team* and fulfill requests from other SPARRING PARTNERS, as defined in Sec. E.3.2. The exercise spans two days. During the first day, the TRAINEES familiarize themselves with the virtual environment. The second day is devoted to the actual training session, which lasts 6 hours. A brief (up to 30 minutes) personalized feedback session follows right after the exercise. Finally, there is another feedback session approximately two weeks later, in which organizers elaborate on the strengths and weaknesses of each team. From each exercise, we collect network flows, computer logs, user commands, and automatic and manual scoring records.

The variability and complexity of CDXs are substantially bigger than in CTFs. The preparation of a new training run of Cyber Czech exercise takes tens of person-months. A unique training definition is created almost from scratch each year and is only repeated a few times. Only a GUI for the *execution* and *reflection* phases are currently supported in the KYPO Portal, both to a limited extent.

The *technical scenario (D_1)* is comprised of the infrastructure of nearly 200 computer nodes in multiple local networks, scheduled attacks and respective vulnerabilities, and configuration of monitoring tools for both trainees and organizers. Multiple iterations make the preparation very laborious. Each Cyber Czech exercise series is framed with a unique story and additional non-technical tasks. The *assessment criteria (D_2)* include several dozen automatically scored network services (e.g., availability of web server or database) and up to 30 manually scored tasks (e.g., penalties for individual attacks, communication with the

sparring partners from the *white* team or *fictitious users*), and requests for reverting malfunctioned network nodes. Complex dependencies in which one network service (e.g., active directory) depends on other services (such as DNS) often exist. All this complicates the design and implementation of a unified data scheme and corresponding front-end tools. Correctness and the estimation of difficulty of training definitions are addressed by so-called "dry runs" in which the whole exercise is tested by volunteers. However, the approach is costly and can be misleading because the readiness of testers may significantly differ from the readiness of target learners.

E.6.2 Visual Analytics of Capture the Flag Games

Insight of Trainees (V_1). Trainees gain insight into the game content through the web-based KYPO portal, which provides them with task descriptions, hints, and solutions for each level and also shows information about the current level and remaining time of the training session. The *Network Topology* visualization (Fig. E.6.1) mediates remote access to individual hosts via a web browser and provides situational awareness by decorating a simple network graph with various semantic symbols. For example, it is possible to support V_{1A} by coloring network links depending on current throughput, and V_{1B} by glyphs distinguishing logical roles of hosts (attacker, victim), or events captured in hosts (e.g., received mails). The importance and quantity of this semantic data differ between training definitions, and they also vary in time. Combining them meaningfully and showing them at the right time so that the trainees are not overburdened is a challenging task.

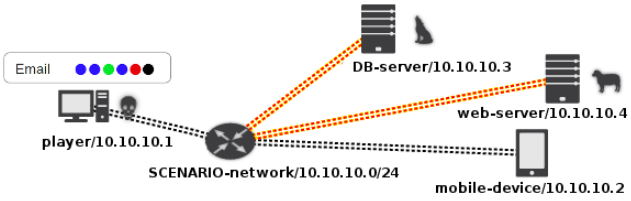

Figure E.6.1: Network Topology with glyphs supporting situational awareness.[1]

Insight of Organizing Participants (V_2). Since we currently support attack-only CTFs without sparring partners, no special visualizations for V_{2A} exist in KYPO.

Supervisors use *CTF Training Session Overview* visualization (Fig. E.6.2) that displays the progress of trainees throughout the CTF game. Each row captures the training session of individual trainees, who can start at slightly different times. Colored bars represent levels. Dots represent user events (e.g., taking a hint), vertical lines show expected level duration. Supervisors use this view to actively manage the training session (V_{2B}) by looking for trainees in trouble (e.g., those stuck in a level for too long, those repeatedly

[1] We provide a full-page version of the visualization in Supplementary Materials at `https://www.kypo.cz/media/3197111/tvcg19-supplemental-materials.pdf`

trying to guess the flag to pass the level instead of solving the task, or those about to quit without trying, which is signaled by displaying all the hints and the solution shortly after each other).

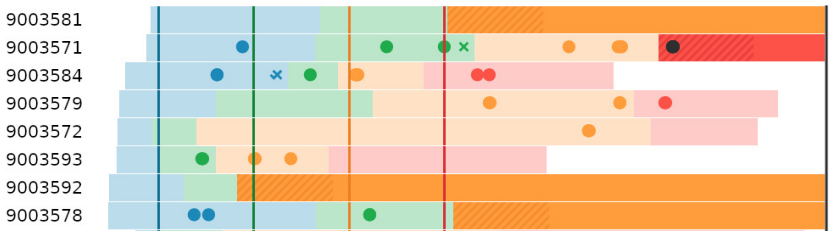

Figure E.6.2: CTF Training Session Overview shows the progress of individual trainees during the training session.[1]

Since our CTFs are executed in the complex cloud-based KYPO Cyber Range, dealing with technical issues is delegated to specialized *operators* managing this infrastructure. They gain insight into the infrastructure state (V_{2C}) via off-the-shelf OpenNebula Sunstone dashboard (see supplemental materials[1]).

Personal Feedback (V_3). At the end of a session, TRAINEES receive a *CTF Feedback Dashboard* [179] supporting V_{3A} with two complementary views (Fig. E.6.3). The left view provides the final score overview for comparison with other TRAINEES. The lengths of the bars show the time of the slowest trainee; different color intensity provides information about the average time. The right side of the dashboard displays the individual score development in time throughout the game. The width of striped areas represents time spent in levels. Dots represent user events. A very similar dashboard is used by SUPERVISORS (V_{3B}) who, in addition, can plot multiple TRAINEES into the score development time series chart for comparison.

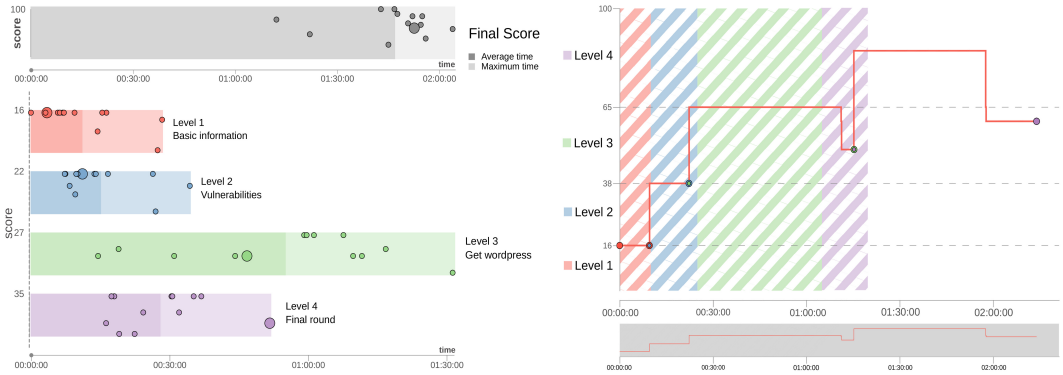

Figure E.6.3: CTF Feedback Dashboard providing individual view on TRAINEE's score results and development in time.[1]

Quality of Training Exercise (V_4). Qualitative aspects of CTF *training definitions*

are supported in KYPO by simple statistical visualizations, e.g., histograms and boxplots capturing the distribution of scores gained by trainees. The *CTF Feedback Dashboard* (Fig. E.6.3) from *personal feedback (V_3)* can be also used to identify weak parts of the training, e.g. levels where trainees spend a long time. However, deeper research and the design of narrowly focused visualizations for quality-related analysis is a future work opportunity.

Behavior Analysis ($\mathbf{V_5}$). Behavior in connection with cybersecurity is often linked to attack graphs and estimation of weak points in the network. A study [20] introduced a method for analyzing computer network security. The method operates with attack paths that represent a linkage of individual nodes with conditions of compromised network security. The output is an attack graph with behavior prediction, and the authors propose the use of their method for incident response training. As for CTF games, the method could also bring insight to the trainee's actions and help the instructor to monitor progress or strategies.

Infrastructure Analysis ($\mathbf{V_6}$). The already mentioned off-the-shelf dashboard provided by OpenNebula Sunstone is currently used also for the basic qualitative evaluation of the underlying cloud infrastructure of the KYPO Cyber Range. However, its utilization for these tasks is not very effective, as it is a universal cloud management tool.

E.6.3 Visual Analytics of Cyber Czech

Insight of Trainees ($\mathbf{V_1}$). Since Cyber Czech is mainly a technical exercise, awareness of the network state V_{1A} and cybersecurity posture V_{1B} are intentionally restricted to resemble real-world settings, as discussed in Section E.5. Trainees interact with a network topology visualization similar to Fig. E.6.1. However, the network infrastructure is more complex, and there are no semantic decorations. Instead, the trainees use a standard monitoring tool (Nagios) showing the status of the network services they are trying to protect. Further, they can infer the consequences of their actions only from the real-time *CDX Scoreboard* (Fig. E.6.4) displayed during the exercise. The scoreboard shows the current total score as well as per-category scores and penalties of all *blue teams*, allowing them to compare themselves. The use of a restricted table-based view is intentional, as we aim to simulate real conditions during the CDX with only limited real-time feedback.

Insight of Organizing Participants ($\mathbf{V_2}$). *Training progression (V_{2A})* of the *red team* is supported by *CDX Attack Plan* (Fig. E.6.5) showing the interactive plan of individual attacks and their state (inactive/ongoing/completed). The green color stands for successful attacks; red stands for unsuccessful ones (i.e., the *blue team* has defended themselves). Attack type abbreviations and given penalty points are shown within each block. Clicking on an attack block reveals further details (e.g., additional comments or screenshots). The *green team* uses the *Nagios* service monitoring system to watch the infrastructure (V_{2C}), to detect when the trainees (un)intentionally blocked some of the monitored and scored services, and to provide brief advice (V_{2B}). Visual insight of other organizing participants

Cyber Exercise Score

Team Name	Services	Attacks	Injects	Users	VNC	Total Score
Blue Team 1	91,843	-8,500	9,000	-1,100	0	**91,243**
Blue Team 4	74,518	-11,000	6,650	0	-4,000	**66,168**
Blue Team 3	85,756	-12,000	2,475	-1,700	-9,500	**65,031**

Figure E.6.4: CDX Scoreboard shows the current scores of all *blue teams*.[1]

is not currently supported.

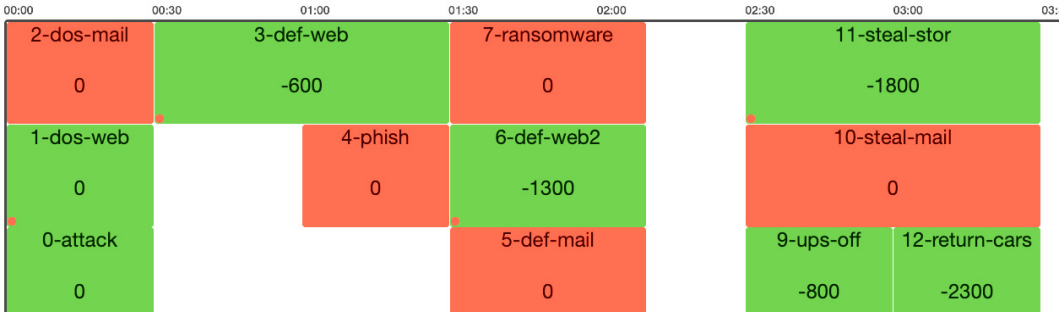

Figure E.6.5: CDX Attack Plan displays scheduled attacks of the *red team* at the end of a 6-hour long training session.[1]

Personal Feedback (V_3). During the hot-washup session, organizers give immediate verbal feedback to TRAINEES. *Personal reflections on the trainees (V_{3A})* are supported by presenting them the *CDX Attack Plan* (Fig. E.6.5) that was hidden from the TRAINEES during the exercise. TRAINEES are also provided with the *CDX Personalized Feedback* [242] (Fig. E.6.6) that shows the score development of their *blue team*. Dots include details about penalties entered by *red*, *white*, and *green teams*. Each dot is associated with a short feedback poll used for gathering further information from TRAINEES. The data is used in the follow-up analysis. The *impact of supervision* V_{3B} is not currently supported.

Quality of Training Exercise (V_4). Vorobkalov and Kamaev [238] describe an approach to the quality estimation of e-learning systems. Their learning process model is based on an extended stochastic Petri net. The method has been implemented in an automated system, and it focuses on helping the expert to perform e-learning process analysis and to deduce learning course mistakes. However, it covers only systems based on net models. For CDX training, the model would not reflect the closely related state of the operational environment. Furthermore, when we consider the unstructured nature of CDX, the model would have to be very sophisticated and extensive.

Behavior Analysis (V_5). The above-mentioned method by Bassett and Gabriel [20]

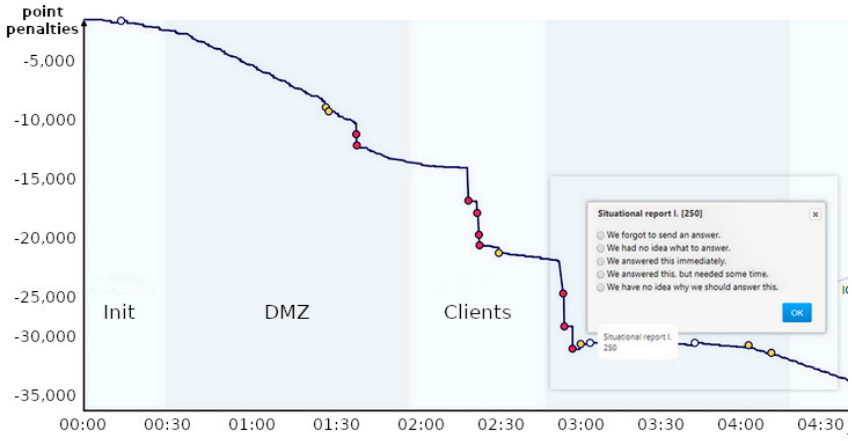

Figure E.6.6: CDX Personalized Feedback shows the score development throughout the training session of a single blue team.[1]

can also be applied to the CDX use case. In this embodiment, the method could be utilized in the form of an attack tool to execute or simulate the events and conditions in the attack graph. The trainee would then receive the output, helping them identify attacks they were facing and allowing them to learn from the events retrospectively (since in CDX, we don't usually want to give them any instant feedback). However, such output would have to be further transformed into a visual form suitable for this type of training.

Infrastructure Analysis (V_6). The support for this type of visual analysis is essentially non-existent at the moment. Although the KYPO platform collects some types of relevant data (e.g., system logs and commands entered by blue teams at individual network nodes), the data is processed ad-hoc and manually or not at all. This is usually done for a debriefing meeting of the organizing participants about a week after the training session. The attendees summarize their observations backed by collected data (e.g., feedback forms from the TRAINEES, analysis of the score development). To support the discussion, we are developing an analytical tool for CDX evaluation that will provide a timeline visualization of automatic and manual logs together with the communication threads among the *blue team* and corresponding *white team* members (Fig. E.7.1).

E.7 Discussion

In this section, we emphasize four key observations we attained and present the challenges for future visualization research in the domain.

The current visualization tools support only situational awareness during the execution phase. The main focus of training sessions is on the execution phase. Therefore, visualizations are designed to provide insight both to trainees (V_1) and organizing participants

(V_2). The reflection phase, in contrast, is vastly unsupported, with the exception of personal feedback (V_3) for trainees.

Organizers have limited insight into the educational impact on learners. The design of cybersecurity training sessions is driven mainly by technical aspects. Training sessions often aim at mastering a particular cybersecurity technique or procedure without focusing on broader learning goals. To overcome this issue, the top-down approach of designing the training must be applied, starting from defining learning goals and going down to a selection of particular techniques. Visual measuring and comparing the quality of learned skills, which is largely overlooked, could help in this process. There is a broad unexplored research area in training quality (V_4) and behavior (V_5) analysis.

Organizers underestimate infrastructure monitoring and analysis. CTF and CDX depend heavily on customized monitoring and management tools for the underlying infrastructure (V_{2C}). However, these tools are lacking. Low-level monitoring tools and other general-purpose solutions, which do not provide a complex overview of the situation, are preferred to customized ones. Analytical tools for post-event infrastructure analysis (V_6) are also lacking.

Data collection is not a problem; data processing is. It is possible to collect large amounts of multivariate data either from the emulated network environment (e.g., network flows, computer logs, commands entered) or from the user interfaces of the cyber range (e.g., mouse tracking, and clicks). The bottleneck lies in data processing and presentation, as we point out in the demonstrative examples. Especially in CDX, data correlation is a difficult task. With rising interest in the quality of training exercise (V_4), a behavior analysis (V_5) could accelerate the demands on the use of the data.

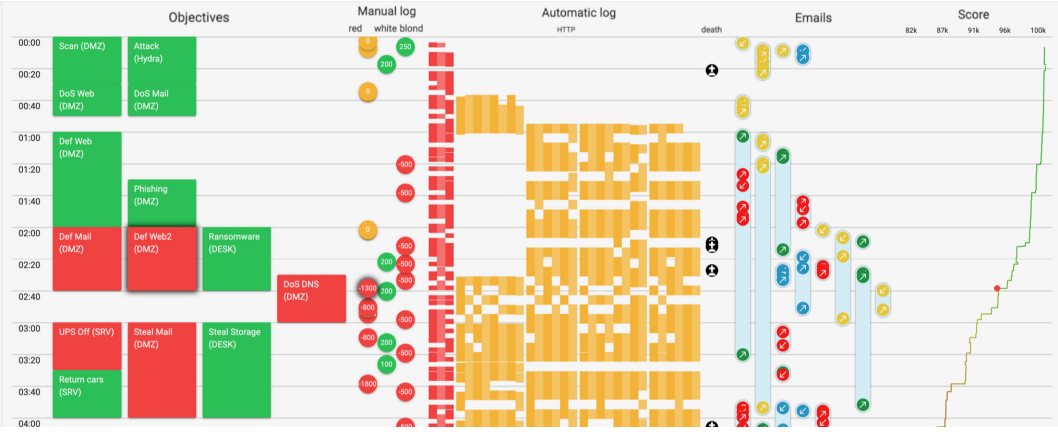

Figure E.7.1: Prototype of CDX Analytical Dashboard.[1]

Challenges for the visualization community are a reflection of the absence of tools. Table E.7.1 summarizes users who benefit from the six visualization tasks, as revealed by the conceptual model in Section E.5. Each bullet represents a visually-analytical use case.

Table E.7.1: The mapping of the low-level roles on the visualization tasks.

	trainee	sparring partner	supervisor	designer	operator
V_1	•				
V_2		•	•		•
V_3	•		•		
V_4				•	
V_5	•	•	•	•	•
V_6				•	•

However, only a few use cases are somehow covered in current practice. For the post-exercise analysis, the main challenge is to find meaningful uses of the collected data to improve the SUPERVISORS' understanding of TRAINEES skill development as well as to provide insight into the training processes for DESIGNERS. Another challenge is to design and develop VA tools to help the DESIGNERS and ORGANIZERS test their hypotheses. Last but not least, it is necessary to revisit the tools for situational awareness of participants during the exercise and provide them with timely individual feedback.

E.8 Conclusion and Future Work

Hands-on cybersecurity training is crucial in educating the future workforce. However, measuring the effectiveness of the training process, using either technical or educational indicators, remains largely unexplored. Our work is motivated by a desire to improve these aspects by applying visual analytics. To the best of our knowledge, this paper is the first attempt to describe the application of VA models to hands-on cybersecurity education.

We used software engineering methods to describe the training life cycle and formalize user roles involved in cybersecurity training sessions. The foundations of our work lie in the existing generic VA models. We systematized the visualizations and hypotheses into six categories and demonstrated the application of the VA model on two classes of cybersecurity training hosted at the KYPO Cyber Range platform. The main limitation is the lack of details from other cyber ranges and training sessions. However, we assume that they are on a similar level of maturity. We back this claim with the experience of our university cybersecurity team members from their participation in events similar to the Cyber Czech exercise series.

Each of the six visualization tasks of the presented conceptual model deserves further investigation. The definition of specific guidelines that can help VA designers and researchers build visual tools is out of the scope of this paper. However, this paper aims to serve as a framework for such guidelines, providing researchers relevant use cases where the application of VA is demanding. We hope that our work will help to establish the agenda for advancing the state of the art and motivate other visualization researchers to explore the domain in

which tehe research areas of education, cybersecurity, and data visualization intersect.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822). Computational resources were provided by the European Regional Development Fund Project CERIT Scientific Cloud (No. CZ.02.1.01/0.0/0.0/16_013/0001802).

Article F

Data-driven insight into the puzzle-based cybersecurity training

Karolína Dočkalová Burská[1], Vít Rusňák[2], Radek Ošlejšek[1]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2] Masaryk University, Institute of Computer Science, Brno, Czech Republic

Computers & Graphics. To appear, 13 pp.

Abstract

Puzzle-based training is a common type of hands-on activity accompanying formal and informal cybersecurity education, much like programming or other IT skills. However, there is a lack of tools to help the educators with the post-training data analysis.

Through a visualization design study, we designed the Training Analysis Tool that supports learning analysis of a single hands-on session. It allows an in-depth trainee comparison and enables the identification of flaws in puzzle assignments. We also performed a qualitative evaluation with cybersecurity experts and students. The participants apprised the positive influence of the tool on their workflows. Our insights and recommendations could aid the design of future tools supporting educators, even beyond cyber security.

F.1 Introduction

Higher-order thinking has become one of the essential skills for the 21^{st} century. The best way to develop and strengthen these abilities is through practical hands-on courses [157, 155]. One commonly used learning method for training problem-solving or various IT skills (e.g., programming) is puzzle-based learning. Michalewicz et al. [162] introduced a game-based

learning method that uses puzzles as a metaphor for getting students to think about how to frame and solve unstructured problems. In IT education, the puzzle-based learning approach has been prevalent for many years [258, 159, 104]. Even programming courses consist of basic concepts such as *recursion* with assignments like "Write a program to calculate the factorial of a given number."

Multiple studies confirmed the usefulness of puzzle-based learning also for cybersecurity education [92, 107, 58]. However, while hands-on training produces a tangible output in many learning areas, e.g., a code that can be checked, analyzed, and evaluated, cybersecurity training is process-oriented. Puzzles are tasks like "search for a vulnerability on server X" that are difficult to track. Tutors have only a limited view of what trainees are doing in the computer network and how they deal with the task, making the post-training evaluation challenging. This paper presents results of cooperation with cybersecurity education experts that led to the design of a visualization tool supporting the follow-up learning analysis of the training sessions.

Regardless of the education subject, tutors make intensive efforts to create, organize, and continually improve these so-called *blended courses*[4]. Trainees' assessment, which usually follows the training session, is integral to the teaching process. The focus lies on comparing individual trainees and analyzing their progress or discovering weaknesses in the training design.

We contribute to the state of the art of applying visualizations in education practice with: (a) a user requirement definition on support tools for tutors of the hands-on puzzle-based learning activities (in the cybersecurity education context); (b) design and implementation of the visualization tool for the post hoc analysis of data from the training session; and (c) an evaluation with domain experts resulting in design recommendations for future work.

F.2 Related work

Assessing the effectiveness of game-based learning poses a significant challenge in the learning analytics research domain. Loh [141] distinguishes between "assessment *for* learning" and "assessment *of* learning." The former is designed to assess a learner's understanding at the course end. The latter is more helpful to educators because it helps them to improve the learning processes. This paper deals with educators' insight into the learning process. A considerable effort has been made in the past to conceptualize data mining and digital assessment for serious games so that generic learning analytics principles can be researched and applied regardless of the specific game content [52, 12, 180]. Our solution deals with event logs and the score-based assessment that represent broadly accepted types of telemetry and evaluation data for serious games.

[4] Blended courses combine computer-supported learning activities with traditional face-to-face interaction during training sessions.

Our work lies at the intersection of education, visualization, and HCI research. According to the classification provided in [178], this paper addresses visual data analysis tasks of organizing participants (referred to as *tutors*). Using information technologies in blended courses enables us to collect metadata produced by learners. Tutors can use them for a post hoc analysis of learners' progression and content revision. Nevertheless, the design and deployment of efficient support tools remain a challenging problem [193]. There are general tools that could be used for specific post-training tasks, e.g., comparing score-based assessment settings via the LineUp application [96]. Our tool aims to reflect the well-defined requirements of training designers and tutors, providing them with a domain-specific comprehensible analytical dashboard.

The purpose of the post-training learning analysis is to understand and optimize learning processes. Previous works [151, 120, 182, 61, 142] address using visual dashboards for learning analysis and confirm the need for insight exceeding simple summative feedback [145]. Apart from focusing on the learning process, learning analytics in higher education also provide valuable teaching or research resources [217]. Analytical tools can support decision-making and improve pedagogical approaches.

Most of these learning analysis tools focus on the high-level perspective evaluation of students' performance. Existing surveys overview and analyze learning dashboards either for tutors [235, 234, 209] or students [30]. Most of them are related to the uptake of massive online open courses. These tools focus on visualizing learning activity, tracking specific learning goals, and providing a high-level perspective on learners' progress. Moodleboard [224] is a decision support tool for pedagogical engineers and administrators providing both course statistics and detection of flaws or misuses for an open-source learning management system Moodle. LISSA [48] aims at improving student-advisor dialogue during face-to-face consultations. The tool provides an overview of study progress or peer comparison among multiple students. SAM [94] is a general-purpose web-based environment visualizing learners' activities, improving awareness, and supporting self-reflection. Such high-level tools represent domain-independent systems to gather, process, and report the collected and derived data while overlooking disciplinary knowledge practices.

In contrast, tools for lower-level data analysis from practical courses often require considering insight from domain experts because the input data driving the analytical tools are domain-specific. Examples can be found for math [118], where the system tackles the understanding of selected math functions, programming tools [87] that utilize compilation processes and software quality metrics for assessment, or penetration testing [82] based on knowledge graphs. Figure F.2.1 categorizes these tools in two axes: x-axis – single or multiple training sessions; y-axis – data specificity, i.e., from the domain-specific data to derived data and metadata.

We propose the *Training Analysis Tool* (TAT) – a dashboard-like tool for tutors providing data-driven insight into a training session through several linked visualizations. The TAT supports tutors in low-level learning analytics tasks such as inspection and comparison of

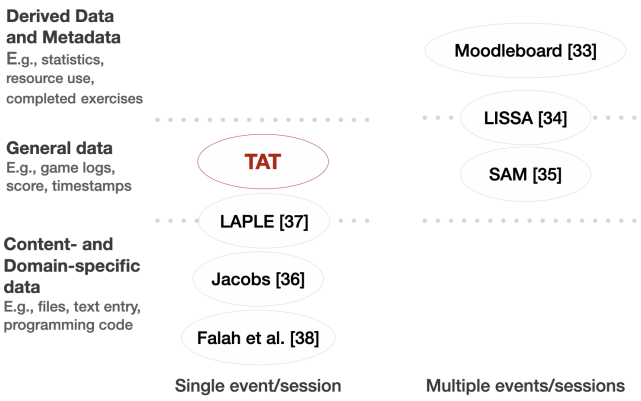

Figure F.2.1: Categorization of learning analytics tools based on their focus (on single or multiple sessions) and the input data types (from domain-specific to derived meta-data). TAT position is highlighted.

trainees or identifying training design flaws based on the data from single training sessions.

F.3 Background

The puzzle-based learning in the cybersecurity domain is primarily represented by *Capture the Flag* (CTF) games [251, 59, 239]. CTF training scenarios serve as puzzle-based templates structuring the content into levels focused on solving cybersecurity tasks, e.g., scan the network, identify a server, find the server vulnerability, exploit it, and gain the root privileges. CTF games can be organized in diverse ways. Very popular are unsupervised online games when a trainee can access the game or interrupt it anytime. Tutored (or supervised) training sessions for small groups are often practiced in a formal cybersecurity education or professional training. The supervised training sessions share the principles of blended courses popular in primary and secondary education.

CTF games contain a short background story, task assignments, their evaluation, hints, and solutions for each level. A typical scenario consists of up to ten levels. Finding a level solution is necessary to proceed to the next one. Training scenarios use multiple gamification characteristics such as scoring, level-based approach, or scoreboards. Trainees are penalized when taking hints or solutions and reach score points for successful solutions.

Hands-on cybersecurity training is often organized in so-called cyber ranges. The *KYPO Cyber Range Platform*[5] (hereafter referred to as *KYPO CRP*)that we use for development and evaluation is a cloud-based environment providing features for the virtualization of computer systems and networks [263]. It serves as a platform for practical training of various cybersecurity skills in university courses as well as for the training of practitioners

[5] `https://kypo.cz`

from institutions outside. The *KYPO Cyber Range* allows us to create so-called *sandboxes* – isolated computer networks consisting of multiple virtual machines for several dozens of trainees (the exact number depends on the cloud capacity and resource requirements). The web portal provides a user interface for the management of sandboxes, users, *training scenarios*, and organizing training sessions.

A typical training session is organized for 15–20 participants in the IT classroom. Trainees log in to the web portal and launch a training scenario consisting of a sequence of cybersecurity puzzles. Trainees solve the puzzles individually in their private sandboxes without affecting others' work. A successful solution of the puzzle yields a short string (called *flag*). Entering the flag in the web portal opens the next level. Trainees who are struggling can use hints specific for each level. When helpless, they can see the correct solution (a list of steps leading to the flag). Time for solving all the levels is usually limited to the class length (one or two hours). Tutors walk around and help trainees either on request or when they realize that someone significantly lacks behind (typically by quick peek on their displays or asking them directly). In the end, the scoreboard shows individual scores, and tutors hold a short debriefing to present correct solutions.

Figure F.3.1 illustrates the principal elements and actions of the whole workflow.

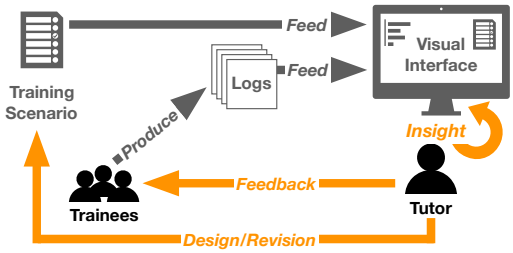

Figure F.3.1: The generalized training workflow. The tutor uses the visual interface to get insight into the training session (to help trainees in trouble) and to revise and improve the training scenario. The data sources are activity logs of the trainees and training scenario description which provides context.

There are two broad use cases for the post-training analysis: (a) a comparison of trainees and (b) training scenario improvements. The former is essential when the CTF games are part of the competitions or exams. The rank or grade is then based on the final score and time. However, the tutor cannot understand the subtle difference in the trainee's behavior or expose cheating. Likewise, *training scenario improvements* were usually based on error-prone manual processing of the logged data and anecdotal evidence from training sessions, making revisions inefficient.

F.3.1 Data description

Hands-on CTF games provide two datasets available for visual analysis: a *training scenario* and timestamped *trainees' events* recorded during the training session. The *KYPO CRP* provides REST API to access these data on-demand in JSON format.

The **training scenario** contains attributes related to the content. Namely, a background story, puzzle assignments, hints, hint penalties, solutions, solution penalties, correct flags, flag score points, and level time limits. These attributes do not change during the training session. However, tutors might edit them afterward based on trainees' feedback or outcomes from training session analysis. Typical changes include fixing typos and improving the clarity of puzzle assignments, or adjusting level duration estimate, score, and penalty points.

The **trainees' events** are automatically collected when trainees interact with the web portal. Example events are: training started, training ended, level started, level ended, correct flag entered, incorrect flag entered, hint taken, solution taken. Each event contains a standard set of attributes (timestamp, event type, training description ID, training session ID, user ID). Three event types (an incorrect flag entered, a hint used, a solution displayed) contain specific attributes – an incorrect flag string and penalty points.

Although the input data is domain-specific, we can find similarities also in other forms of puzzle-based gaming. Data types are either integers (score and penalty points, level duration estimate – representing minutes) or text strings (plain-text for flags, markdown markup for all the rest).

F.4 Process and methods

We closely collaborate with domain experts (cybersecurity educators) from our university who represent target users. They provided initial requirements, gave us feedback on proposed designs, and participated in both evaluations. Our goal was to improve the workflow of tutors and organizers of hands-on cybersecurity training sessions through the design and deployment of the Training Analysis Tool that processes data from the *KYPO Cyber Range*.

In this project, we applied the user-centered approach guided by the design study methodology framework [211], reflecting its *core* stages: discover, design, implement, deploy. Our iterative process has four phases. Each phase reflects one or more of these stages:

Problem characterization *(discover)*: We conducted semi-structured interviews with three domain experts from the university cybersecurity team. All of them partake in educational activities as seminar tutors or lecturers, and they also participated later on in the evaluation. Each interview lasted about an hour. We also did four field observations during training sessions to gather user requirements and complement our notes, each lasting up to two hours. From these data, we elicited functional requirements and design decisions for

both tools.

Early prototype and formative evaluation *(design, implement, deploy)*: We created the early prototype and performed a qualitative formative evaluation with five collaborating cybersecurity educators and one student familiar with the CTF games.

Late prototype and summative evaluation *(design, implement, deploy)*: We added new features and redesigned the user interface based on received feedback. A qualitative summative evaluation with eight participants served us for the validation of the final designs.

Final deployment *(implement, deploy)*: The last phase includes the integration of TAT into the *KYPO CRP*. We also plan to collect further feedback from its routine usage. Unfortunately, due to the COVID-19 pandemic, the number of training sessions has been severely limited.

F.5 User requirements

Post-training session evaluation provides many opportunities for tutors to perform a detailed analysis of a training scenario and assessment of the trainees. The interviews and field observations revealed that tutors struggle with analyzing the training data from the individual sessions. They expressed the need for an overview of the data collected during the training session, which enables them to: analyze trainees' behavior, compare their performance, and revise the content and configuration of the training scenario.

We organized the requirements into the four main categories:

R1 – Trainee behavior analysis: Tutors should examine trainees' behavior and identify outliers – e.g., those who are extremely slow/fast or gave up the training. They should assess the trainees by comparing their results (e.g., final time and score, taken hints, number of entered incorrect flags). It is also relevant when the training session is a part of some competition. Further, reviewing the trainees' actions, such as many partially correct flags submitted by several trainees, can point out flaws in the puzzle assignment.

R2 – Assessment revision: Correctly set scores and penalties are crucial for the gameplay and trainees' motivation to complete the training. Setting the penalties for hints too small, for instance, can demotivate trainees in attempting to find the solution by themselves. Instead, they could take all hints immediately, which would even result in a better final score. Therefore, the tutors should be able to review the assessment criteria of the training session.

R3 – Timing revision: Proper estimation of time requirements for cybersecurity puzzles is tricky. Short time allocated for a challenging puzzle can delay the whole session, put unnecessary pressure on trainees to take hints early, or force tutors to intervene prematurely. During the interviews, even the most experienced tutors admitted that they

do not have a proper first estimate of mapping puzzle difficulty to time limits. Therefore, tutors should be able to review the time limits of the training session.

R4 – Training content revision: Tutors should be able to analyze problematic parts of the training content to improve its quality iteratively. The trouble can be hidden either in individual puzzles (e.g., unclear puzzle assignment, useless hint) or their interconnection (e.g., the unbalanced difficulty of two successive levels).

F.6 Early design

The main goal of the *Training Analysis Tool (TAT)* is to display data from a single training session in the context of the corresponding training scenario (e.g., puzzle assignments, scoring, timing). The tool is designed as a dashboard combining several linked views. Its design follows principles formulated by Oslejsek et al. [178]:

- Analyze the *impact of tutor's supervision:* The tool consists of temporal views of trainees' actions and the score development at various levels of detail. Tutors can analyze the impact of both individual and class-wide interventions by focusing on the time of intervention.
- Analyze *quality of training exercise:* All views display the score and time limits that form the primary assessment criteria and delimit the training session's difficulty. These visual artifacts help tutors to analyze the quality of training. Moreover, predefined parameters (penalties, time limits, tasks) are available in the dashboard together with run-time data, enabling tutors to reveal possible weaknesses in training scenarios by comparing expected versus actual development.
- Analyze *behavior analysis of trainees:* The training session is captured from several perspectives: temporal view on trainees' activities, a static preview of final results, and detailed dynamic score development. By combining these coordinated views, tutors can interactively analyze individual trainees' behavior, compare them mutually or concerning expected behavior, and visually identify outliers.

The early prototype of the *Training Analysis Tool (TAT)* (Fig. F.6.1) is a web application consisting of three interactive visualizations: TIME-SCORE OVERVIEW, TRAINING OVERVIEW, and INDIVIDUAL TRAINING WALKTHROUGH.

The former two are based on visualizations proposed by [179] for player-centered reflection and CTF game results. Since their input data is similar (timestamped events), we used its core design principles and visual encoding, but our visualizations provide extended interaction capabilities. We further elaborate on the design of individual TAT components in detail.

All three visualizations of the early prototype use a fixed color scheme. The colors were meant to distinguish individual levels of training and were selected in different intensities to

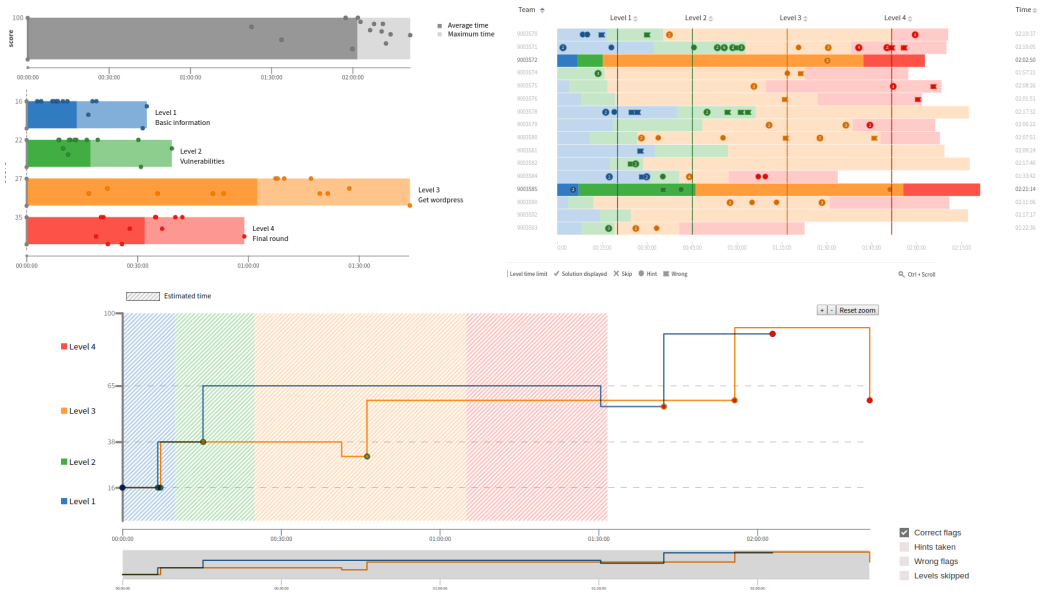

Figure F.6.1: The early prototype of the Training Analysis Tool (TAT) consists of three interconnected visualizations. The TIME-SCORE OVERVIEW (top-left) presents the distribution of achieved scores (final and per-level) for each trainee. The TRAINING OVERVIEW (top-right) displays the overall training duration for each trainee and their activities (e.g., taking hints, inserting incorrect flags). The INDIVIDUAL WALKTHROUGH (bottom) is suitable for a detailed comparison of two or more trainees.

be distinguishable for people with the most common forms of color vision deficiencies.

F.6.1 Time-score overview

Total duration and the final score are two main factors used for measuring the performance of the trainees. The time-score overview (Fig. F.6.1, top-left) helps identify the correlations between these factors, providing a view on the score distribution, pinpoint the outliers, or allocate clusters.

Using simple standard statistical views, such as boxplots, would be inconvenient because we need to put in the context multiple metrics (average, and estimate times, final scores). Therefore, the visualization combines bar charts with scatter plots to incorporate time and score data into a single view. The top bar shows the total time (x-axis) and each trainee's final score (y-axis). The smaller bars below represent individual levels (i.e., tasks). Each bar's length expresses the maximum time for the given level (i.e., the time of the slowest trainee). The average time is on the border of two color shades. Although the scoring span can differ in each level, the bars have fixed heights. The vertical space is sufficient to display and analyze achieved score distribution regardless of the scoring span. The maximal level or game score is on the y-axis, and the exact score numbers are provided on-demand as tooltips of individual dots together with a trainee's name.

Hovering the mouse cursor over the dot highlights the corresponding results of the trainee in the remaining levels highlight and the exact time and the achieved score for the level display. A mouse click on the dot highlights the corresponding data in the training overview and displays detailed score development in the individual training path at the bottom. Dot clusters can visually indicate the correlations between time and score, which is particularly helpful when the tutor aims to identify the training design issues such as a level difficulty compared to its duration.

The tutors can use it to analyze the results of individual trainees and put them in the context of the training group (**R1**) or to review score-based assessment (**R2**). Bar charts also help the tutors review time requirements (**R3**). Dot clusters may help in the identification of problematic levels in the training scenario (**R4**).

F.6.2 Training overview

The training overview (Fig. F.6.1, top-right) provides a detailed yet compact and uncluttered view of the trainees' progressions and activities. It is based on a stacked bar chart where each row corresponds to one trainee. Segments represent training levels and encompass related game events as glyphs. A user can filter the data based on the level duration and zoom the view to unfold the aggregated events (numbered circles) performed quickly.

The visualization shows the relative time of the training. The stacked bars are aligned to

the left, so it is possible to compare the time requirements regardless of the delays caused by individual trainees' various starting times (**R3**). Level labels above the bars support sorting by the duration of the corresponding levels. The related vertical lines indicate the expected level duration. When sorted, they also reveal the deviation of the actual and estimated time for each trainee.

The glyphs indicate events. In this view, they help the tutors to recognize possible problems in the design of training definition (**R4**) or analyze the behavior of the trainees (**R1**). For example, multiple incorrect flags submitted by diverse trainees can indicate unclear or ambiguous instructions; many hints taken in quick succession may suggest a lack of effort caused by improper difficulty.

F.6.3 Individual walkthrough

The INDIVIDUAL WALKTHROUGH (Fig. F.6.1, bottom) is based on a step chart with glyphs representing trainees' actions. It enables the tutors to track outliers' behavior (**R1**) and explore the cause of recognized problems in the training session (**R2**, **R4**). It provides a detailed insight into a trainee's advancement and actions or allows comparing two or three trainees selected from the TRAINING OVERVIEW list or the TIME-SCORE OVERVIEW. The y-axis represents gained score. The horizontal dashed lines imply the maximal level score. The striped background outlines the estimated level times.

A zoom function allows adjusting the view on a selected portion of the chart, which is useful when the events are clustered. On mouse hover, a tooltip shows details for each action. A context view frame below the main chart helps the tutor to get oriented in the zoomed area and shift the time range when needed. Furthermore, the checkboxes in the bottom right corner allow filtering the event types.

F.7 Formative evaluation

The main goal was to gain feedback on the TAT's usefulness in four areas:

- **Trainees** – Is it possible to identify trainees who struggled (e.g., lacking behind, stuck with the task/level)? Can tutors recognize any unusual behavior of trainees (e.g., cheating, prolonged inactivity)?
- **Training session** – Is it possible to recognize when the training is running out of schedule? Can tutors identify scenario design issues?
- **Visual encoding** – Is the visualization easy to understand? What type of information is redundant or missing?
- **Interaction** – How do tutors interact with the visualization? Are the interaction capabilities sufficient?

We further evaluated the usability and usefulness of the visualizations and gathered remarks on visualization improvements for the following design process iteration.

F.7.1 Participants

Due to the necessary background knowledge of hands-on cybersecurity training, we conducted a qualitative user study with five domain experts (P1–P5) and one student (P6). All of them were members of the university cybersecurity team who partake in hands-on training on different positions. Table F.7.1 shows their demographic information.

Table F.7.1: Demographic summary of the participants and their involvement in the design study. TE – teaching experience (in years), OE – organized hands-on exercises (in sessions). Participation in individual stages: PC – problem characterization; FE – formative evaluation; SE – summative evaluation.

ID	Age	Position	TE	OE	PC	FE	SE
P1	33	Lecturer, Manager	4	>20	✓	✓	✓
P2	27	Seminar tutor	7	<20	✓	✓	✓
P3	31	Seminar tutor	3	>20	✓	✓	✓
P4	27	Seminar tutor	5	<10		✓	✓
P5	35	Senior lecturer	5	>20		✓	✓
P6	24	CTF Course graduate	0	1		✓	✓
P7	22	CTF Course graduate	0	1			✓
P8	21	CTF Course graduate	0	0			✓

F.7.2 Procedure

In September 2019, we held the formative evaluation sessions in person using 27" iMac with the resolution 2560×1440 and Google Chrome browser version 76. The experimenter took notes and audio recorded the participants' opinions and thoughts.

The user study had two parts, and the participants were asked to think aloud. The sessions lasted about an hour. In the first part, the experimenter outlined the procedure. The participant consented and filled the demography questionnaire. The experimenter presented the TAT and situated the participant in the role of a tutor using the tool. Next, the participant spent 2–3 minutes familiarizing with it using dummy data followed by completing three tasks addressing requirements R1–R4:

- *T1: Identify an unusual behavior of trainees and name the potential issues.*
- *T2: Find and compare a pair of trainees who: a) have the same score; b) were the best and the worst; c) were the slowest and the fastest. How do they differ?*
- *T3: Identify problems caused by the poor design of the training scenario and propose improvements.*

Participants performed the tasks on two data sets DS1 and DS2. We chose the genuine data since they contain various actions observable during training sessions (e.g., guessing the correct flag, prolonged inactivity, varying trainees' performance). Their different size, number of trainees, and duration show two distinct yet ordinary real-world circumstances.

DS1 is from the tutorial on computer forensic skills and consists of six game levels. The goal is to identify and examine malicious software running in the computer system. The trainees learn how to identify a suspicious application, dissect its executable, and process memory. The session lasted 55 minutes, and 16 trainees generated 374 events, making the 23.4 events per trainee on average. DS2 is an attack-oriented training scenario that consists of four game levels with the following puzzles: exploit server vulnerability, gain the root privileges, access a protected data file, and cover the traces after the attack. Six trainees generated 146 events over 90 minutes, averaging 24.8 events per trainee.

Finally, the participant filled two usability questionnaires and was debriefed. We chose the SUS – System Usability Scale [202] and the SEQ – Single Ease Question [203], two widely used questionnaires for measuring various products' usability. The former is a widely used method for assessing the usability of the systems. The latter is considered a robust measure to quantify the usability for tasks that are too complex for metrics like task duration time or completion rate[6] and when the number of participants is low, as in our case.

F.7.3 Results

The formative evaluation revealed weaknesses in the early design and helped us understand tutors' work after the training session.

The most acclaimed feature of the TRAINING OVERVIEW visualization is the ability to sort trainees by the time spent at some level and compare them to the estimated level duration (defined in the training scenario). Participants also used the visualization to identify the trainees who significantly exceeded the estimated level duration time.

For most of the participants, the SCORE OVERVIEW visualization was a starting point when solving all the tasks. They used it to identify outlying trainees (P2, P4, P6), to assess the difficulty of each level based on the time/score distribution of trainees (P1, P2, P4, P6), or to compare it with the maximum score per level (P3, P4). P3 also used the score overview to assess the conceptual design of the training scenarios (the first levels should be manageable and short compared to the final ones). Participants lacked information about estimated level duration (P1–P4, P6). P6 wanted even more details, such as medians of time and score for each level.

Participants often used score overview visualization to highlight trainees in training overview and vice versa. Score overview was also often used in T2 as a selector for trainees to

[6]The user responds to a single precisely-worded question ("Overall, how difficult or easy did you find this task?"), using a scale from 1 (Very difficult) to 7 (Very easy).

compare. We did not observe any other extensive mutual use of two or all three visualizations. On the other hand, the INDIVIDUAL TRAINING WALKTHROUGH visualization was generally considered "useful only in a specific case when the training session is organized as a competition to decide the final order of trainees" (P4).

The main complaint (mentioned by all) was the absence of a tabular view showing various details of all trainees such as their final score, scores per level, number of taken hints, or incorrect flags.

Other frequent issues were: the absence of filtering features (P1–P5); a missing overview of the training scenario allowing the users to skim through the texts of tasks, set penalties, and flags (P2–P4, P6); insufficient integration of the visualizations (P1, P2, P4, P5); and the visual encoding (P1, P3, P4, P6) considered by P3 as "disturbing due to many colors without proper meaning."

The SUS score was 65.4 points (out of 100). It corresponds to the *good* rating, according to the adjective ratings [18]. Fig. F.7.1 summarizes the SUS questionnaire responses. With the SEQ score of 6.5 (out of 7), the TAT showed to be well-suited for training design analysis (T3: Identify training design issues.). The two tasks focused on identifying and comparing trainees scored 5 (T1: Unusual behavior of trainees) and 6 (T2: Comparison of trainees).

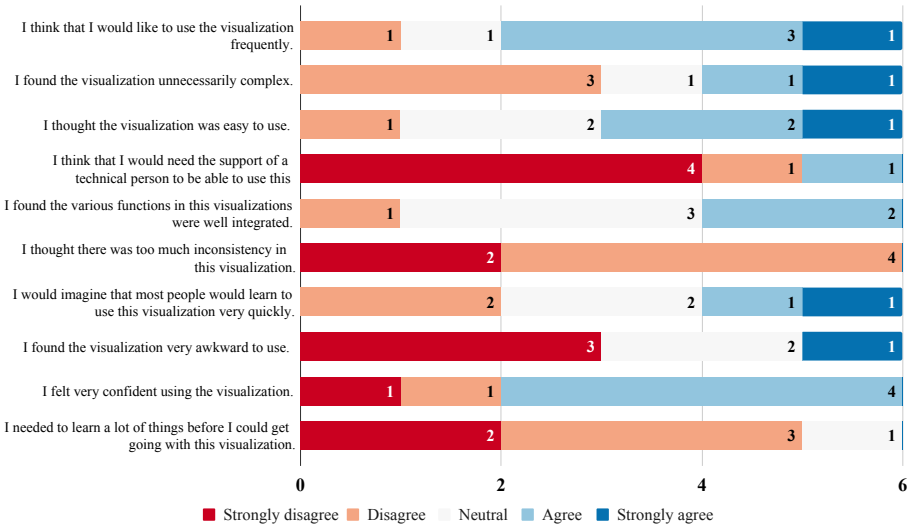

Figure F.7.1: Formative evaluation: The SUS questionnaire responses.

While these results confirmed the overall usability and usefulness of the TAT, we had to address the main issues raised by the study participants.

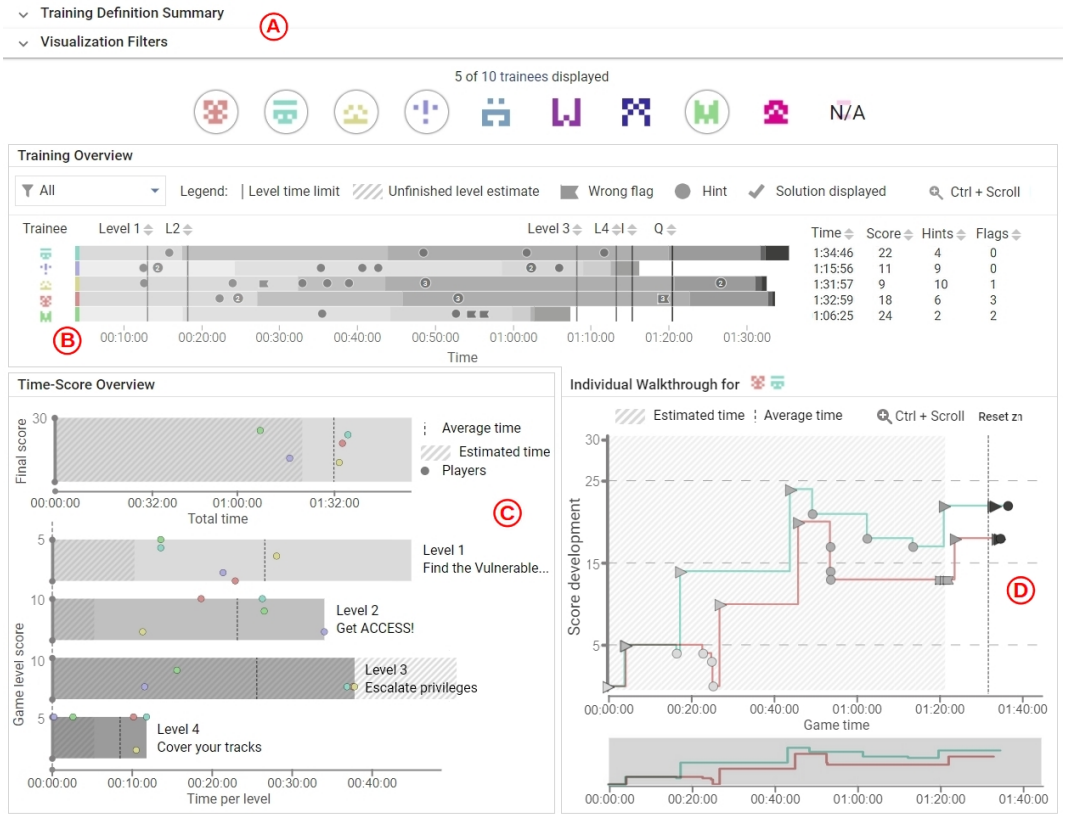

Figure F.7.2: The Training Analysis Tool (TAT) consists of the upper panel for training definition summary and filters (A) and three visualizations: the TRAINING OVERVIEW (B) displays the overall training duration for each trainee and their activities (e.g., taking hints, inserting incorrect flags). The TIME-SCORE OVERVIEW (C) presents the distribution of achieved scores (final and per-level) for each trainee. The INDIVIDUAL WALKTHROUGH (D) directly compares of two or three trainees and is subordinate to the TRAINING OVERVIEW.

F.8 Final design

We revised the final design rationale, visual encoding, and interaction capabilities of the current version of the TAT based on the formative evaluation. The prototype, implemented using Angular and D3.js library, is available at `https://tat.surge.sh`.

The main principles of the three visualizations remain the same. However, we significantly redesigned the layout making the TRAINING OVERVIEW the most prominent visualization. We also added more filtering options for selecting individual trainees and revised the use of colors. The formative evaluation also revealed that the coloring of levels is not essential for the users, so we have changed it in the late prototype: the platform on which the training sessions take place generates a unique avatar for each trainee. Therefore, we

decided to emphasize the trainees based on the avatar's color instead. Now, each trainee has a unique color in all three visualizations. These colors are not intended as the exclusive means of trainee identification but as complementary visual support (to accompany the ability to highlight or filter the trainees). To distinguish training levels, we used gray color shades in the late prototype.

Finally, we added additional information regarding the training definition, such as the task descriptions, correct flags, and contextualized trainees' data with individual levels. Fig. F.8.1 displays the final layout, with the collapsed training definition summary and visualization filters sections.

F.8.1 Training definition summary and visualization filters

The TAT's upper part (Fig. F.7.2 – A) contains a collapsible panel with the training definition details, visualization filters, and avatar-based trainees filter. The training definition summary serves for the configuration of the tool and synopsis of the training. It provides training scenario parameters (i.e., task assignments, hints, penalties, correct flags). The tabs show data for individual levels (Fig. F.8.1 – A). For each game level, a table summarizing data of individual trainees provides an overview of the gained score, taken hints, incorrect flags, and time spent in the level (**R2** and **R3**). Comparing the results shown in the table with the level content and parameters (e.g., the comparison of incorrect flags with the correct flag or scheduled time allocation with the average or median values) can help the tutors identify problematic parts of the content (**R4**).

The Visualization Filters (Fig. F.8.1 – B) are global filtering options to show or hide glyphs representing hints or flags and switch between trainees' avatars and names (IDs). The avatars (Fig. F.8.1 – C) are switches for filtering out the trainees from the training overview and time-score overview.

F.8.2 Training overview

We extended the training overview (Fig. F.7.2 – B) with the table summarizing total game duration, achieved score, number of taken hints, and submitted incorrect flags for each trainee. We also added the legend for quicker orientation.

The training overview interacts with two complementary views. By clicking on the stacked bar, the individual walkthrough visualization appears, showing score polyline and events of the corresponding trainee. The level bars highlight the corresponding dots in the time-score overview and the polyline in the individual walkthrough on mouseover.

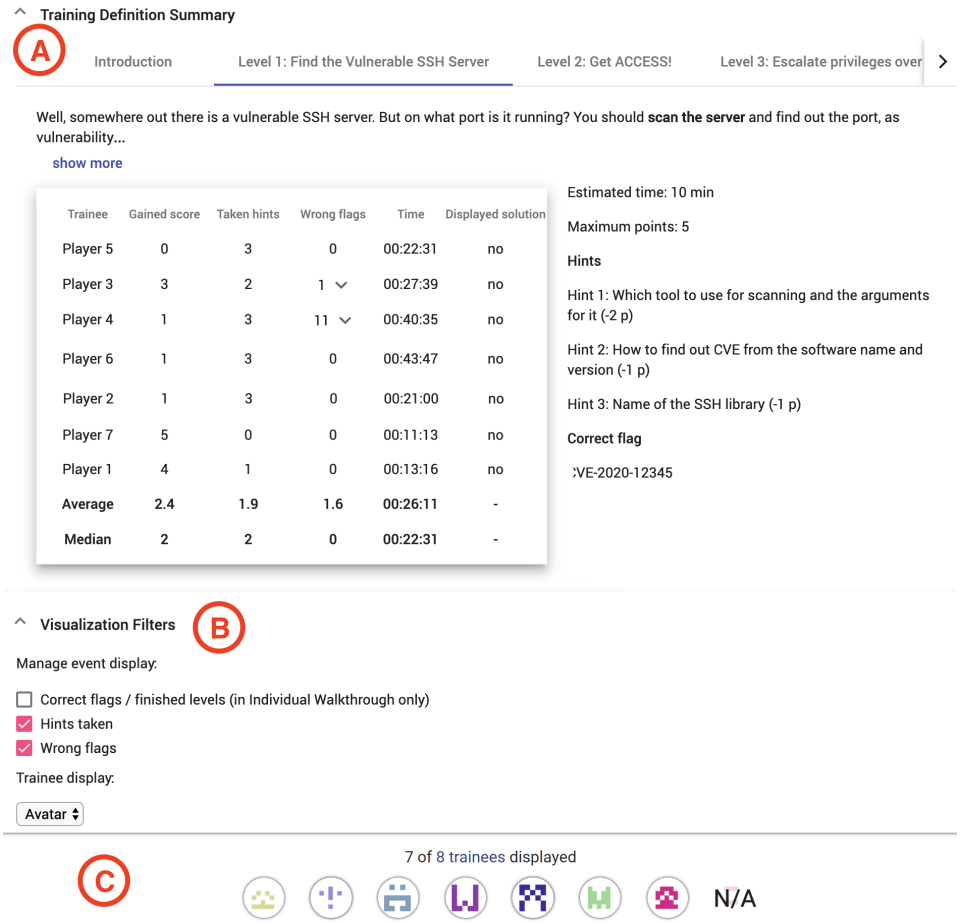

Figure F.8.1: TAT – details of the TRAINING DEFINITION SUMMARY (A), CONFIGURATION (B), and the TRAINEES (C) sections.

Time-score overview

Unlike the early prototype version, we added the dashed vertical line to indicate the actual average completion time of the trainees. The striped segments delimit the time estimate for each level. Therefore, the tutors can quickly identify the differences between the expected and the actual (and averaged) time for each level, as shown in Fig. F.8.2.

Individual walkthrough

In the final version, the INDIVIDUAL WALKTHROUGH (Fig. F.7.2 – D) displays upon selecting a trainee in the TRAINING OVERVIEW. The selected trainees are indicated as avatars next to the title. We also reflected the main complaints regarding the clutteredness and simplified the visualization layout. Only the total training duration estimate is shown instead of the

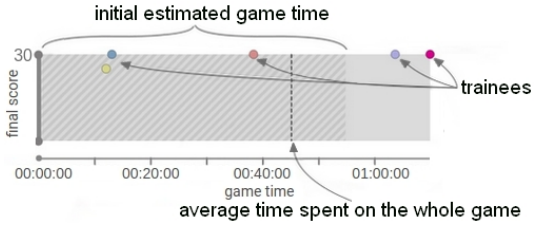

Figure F.8.2: The Time-Score Overview combines bar charts with scatter plots to show relationships between the score and time of the game levels.

estimate for each level. We also added the vertical dashed line to indicate the actual average time, similarly to the time-score overview.

F.9 Summative evaluation

The summative evaluation was held in April 2020. We intended to validate the final design concerning the user requirements **R1**–**R4**, assess the usability and usefulness of the TAT, and identify possible refinements for the final integration into the *KYPO CRP*.

F.9.1 Participants

We asked the same six people who participated in the formative evaluation. We also recruited two more students who passed the CTF design course taught at our university (see Table F.7.1). They represent novice users familiar with CTF games' basic concepts and only have hands-on experience with their design.

F.9.2 Procedure

Due to the COVID-19 pandemic restrictions, we held it remotely using Google Meet, which we also used to record audio and screen. The participants used their computers or laptops with the 13.3"–27" screens and resolutions ranging from FullHD to UHD. The procedure was almost the same as in the formative evaluation (see Sec. F.7.2). The only difference was a new data set that we used for the tasks.

DS3 uses data from a training session held as the introductory lecture of the CTF game design course of Fall 2019. It is an attack-oriented four-level training scenario similar to DS2, in this case, tested on nine trainees who generated 281 events in the session lasting 110 minutes. On average, each participant performed 31.3 events.

F.9.3 Results

The participants completed all the tasks without struggle. Despite minor difficulties, the immediate feedback was more positive than in the previous evaluation. Since the tasks are complex and depend on the tutor's knowledge and experience we sought qualitative input rather than measuring user performance.

Participants mostly worked with the training overview since it contains most of the necessary information. The time-score overview serves well to identify timing issues and assess level difficulty. The training definition summary supports finding flaws in the puzzle assignments (e.g., misleading texts, wrong instructions for flag format). Further, we did an inductive qualitative analysis [226] of the video recordings, which is summarized below.

Visualizations usage. Figure F.9.1 shows the usage of visualizations to solve the tasks by participants. The most preferred was the training overview. All but P5 used the training overview as a starting point when solving all the tasks (P5 preferred the time-score overview). Its most acclaimed feature is the ability to sort trainees by the time spent in individual levels and compare them to the estimated level duration (defined in the training scenario). Participants also used the visualization to identify the trainees who significantly exceeded the estimated level duration time. All the sorting options (by time spent in a level, final time, score, hints, and incorrect flags) were used at least once by each participant. On the other hand, the zooming function was used only rarely (P1, P6). The participants used the time-score overview to identify outlying trainees (P2, P4, P5), assess each level's difficulty based on their time/score distribution (P1, P2, P4, P5), or compare it with the maximum score per level (P3, P4). The individual walkthrough was still considered the least usable (P1, P2, P5, P6, P7). P1 and P5 did not work with it at all. Others used it only for a direct comparison of two trainees (*T2*).

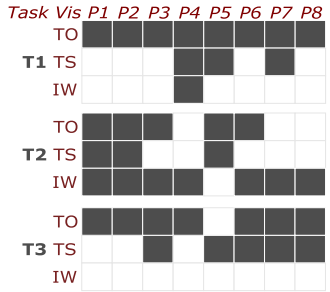

Figure F.9.1: Gray cells indicate visualization usage (Vis) when solving tasks T1–T3 for each participant (P1–P8). Visualizations: Training Overview (TO), Time-Score Overview (TS), Individual Walkthrough (IW).

The TAT allows comparison of trainees beyond time and score. To identify non-standard trainees' behavior (*T1*), we observed that all the participants revealed all or

almost all occurrences of the most common types, such as taking all hints at once shortly after they entered a new level or guessing the flags in each dataset. The participants found those with the lowest score/largest time, followed by a detailed inspection of the number of taken hints and inserted incorrect flags. The procedure was the same for all. The difference was only in the starting visualization. While P4, P5, and P7 started with the TIME-SCORE OVERVIEW, the rest used the TRAINING OVERVIEW solely. The participants also intensively used the trainee filter combined with the TRAINING OVERVIEW sorting capabilities to filter out unwanted trainees quickly, especially for the second task (*T2*). Despite the INDIVIDUAL WALKTHROUGH received mixed reactions, most participants (except P1 and P5) used it for a head-to-head comparison.

The TAT helps to identify training scenario shortcomings. When dealing with the identification of training scenario shortcomings (*T3*), the participants mainly focused on three areas: correcting the time estimates and maximal score of individual levels, the perceived level difficulty, and instructions for a correct flag format. All the participants proposed changing the time estimates or the assigned maximum of points based on the trainees' overdue in the first two levels of D3. Moreover, seniors (P1, P3–P5) also identified the confusion with the flag formatting instructions in the second level. P3–P5 analyzed the data even more profoundly and revealed the flaw in the game design based on the observation that some trainees used the correct flag for the fourth level in the third one.

Except for P1, P2, and P4, the participants used the TRAINING DEFINITION SUMMARY since it clearly shows the difference between the estimate and real-time. The size of each level allows for a quick comparison of their perceived difficulty (the longer it took, the problematic the level was). The glyphs visualizing incorrect flags in the TRAINING OVERVIEW proved to be good indicators for potential issues with the puzzle assignments, including the technical instructions. All the experts (P1–P5) greeted the TRAINING DEFINITION SUMMARY as a convenient way to search for problematic parts of the training definition.

Gaps and drawbacks of the TAT. We received several suggestions for further improvements to the TAT visualizations. P5 suggested adding "the horizontal line also showing the average score per level" in the TIME-SCORE OVERVIEW to improve comparing level scores. The two-level filtering (avatars → trainees in the TRAINING OVERVIEW) received mixed feedback. Only three participants (P1, P2, P5) used both to filter out specific trainees, while others preferred to keep all of them visible. The evaluation also revealed that with the grayscale for the TRAINING OVERVIEW, highlighting of selected trainees is not very pronounced and will be revised in future development.

The main benefit of the INDIVIDUAL WALKTHROUGH is that the polyline visualizing score development better informs the tutor whether there are similarities in the trainees' gameplay. Since this is useful only in a specific use case, we will reconsider its integration in the subsequent design iterations simplifying the user interface.

The average SUS score raised to 77.5 (compared to 65.4 for the early prototype), which

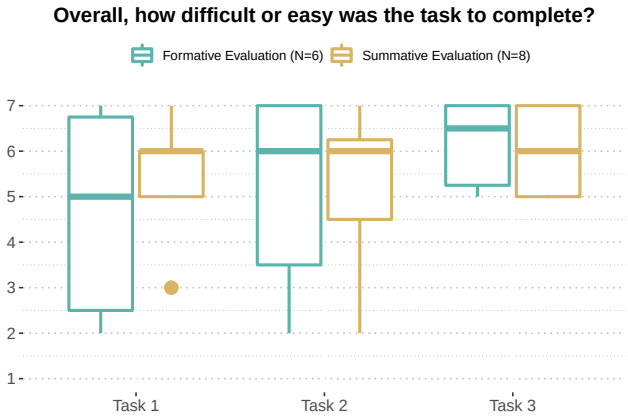

Figure F.9.2: Single Ease Question scores of the tasks in both evaluations.

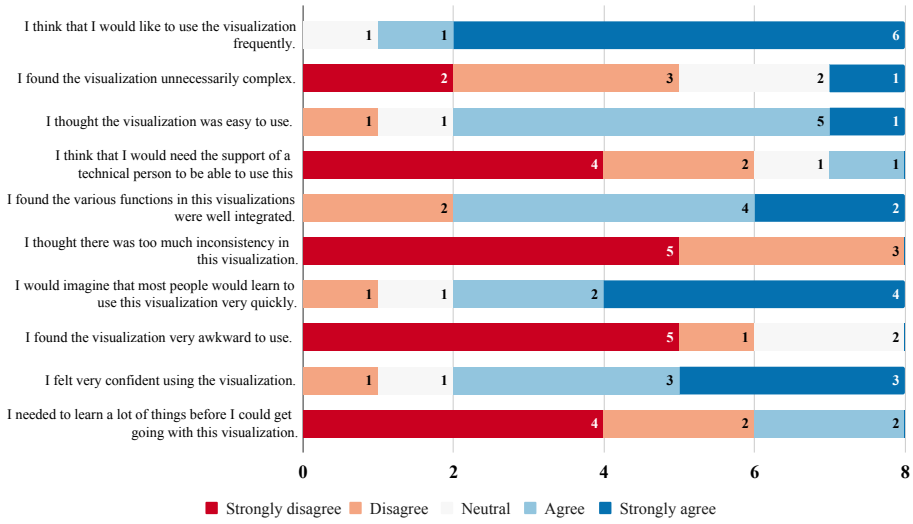

Figure F.9.3: Summative evaluation: The SUS questionnaire responses.

still equals to *good* rating. We assume that it is mainly due to the higher complexity of the tool and the remaining issues with the INDIVIDUAL WALKTHROUGH. The data plot of the SUS questionnaire responses is in Fig. F.9.3. However, the medians 6.0 of SEQ score (Fig. F.9.2) for all the tasks (*T1*–*T3*) further supports our statement that the TAT is well-suited for the post-training analysis.

F.10 Discussion

In this section, we discuss the findings and limitations of the studies.

F.10.1 Lessons learned

The summative evaluation validated our design decisions. The verbal feedback from participants and the SEQ and SUS scores confirmed that the tools address the elicited requirements. We also revealed three notable findings regarding the presentation of summaries, sorting and filtering capabilities, and domain specificity.

Summaries. Extending the visualization with pertinent summary data could help tutors to overview the situation and identify anomalies quickly. Especially in analytical tools, even elementary statistics and simple charts are helpful. Although we did not implement such charts in the TAT, some participants asked for them as feature requests.

Sorting and filtering. The evaluation revealed that we should work with the sorting and filtering options even more thoughtfully so that tutors can better focus their attention. There must be a real usage scenario for each filter type. Particular attention should be paid to carefully selecting items for filtering and the batch selecting and filtering shortcuts (e.g., "deselect all").

Domain-specific insight over universality and scalability. Puzzle-based learning represents a vast area where tutors' support tools differ vastly among various application domains. Since there are no guidelines or best practices and the user requirements are often contradictory, they have to be considered carefully, and the tools should be tailored to specific uses. Furthermore, the amount of data from a single session is usually relatively small.

F.10.2 Limitations

Both user studies had two main limitations to the external validity: low number of participants and qualitative focus of the evaluation in the controlled environment instead of the in-the-wild evaluation.

To ensure the evaluation's ecological validity, we needed users with practical experience with organizing hands-on training sessions and knowledge of cybersecurity education. These demands notably restrict our choice of suitable candidates. Our collaborating cybersecurity educators are, no doubt, the primary users of the developed tools. Therefore, they provided relevant feedback, which will serve as a source for further thoughts on both tools' improvements. We also asked students of the cybersecurity degree program who successfully passed the university course on CTF games design. They represent novice users unfamiliar with analytical visualizations.

Due to the qualitative nature of the evaluations, we did not focus on finding the limits in terms of the total number of trainees and their events since the events with more than 16 participants are literally none due to the space limits of the training facility at our university. We originally planned to perform the case studies to assess the TAT's final design in the

actual deployment. Unfortunately, due to the COVID-19 pandemic, the scheduled hands-on training sessions had been canceled, and the only feasible option was to perform the evaluation remotely, using the same procedure as in the summative evaluation.

In this work, we restrict ourselves to the case study of hands-on cybersecurity courses focused on system hacking and cyber-attacks. In particular, puzzle-based capture the flag games where the structure and data are well-defined in advance. These restrictions allowed us to provide the tutors with a more in-depth insight into this specific application sub-domain through a pair of visualization tools.

Despite these limitations, the provided feedback has been guiding our work and feature requests for the deployment into the *KYPO CRP*.

F.11 Conclusion and future work

We introduced the visual analytics tool that, based on the qualitative feedback, improves the tutors' insight into the training sessions and allows them to assess the quality of the training scenarios and evaluate the training session results. We focused on low-level learning analysis (i.e., analyzing data from a single training session). As we pointed out in Sec. F.2, this particular area is often overlooked since the main focus in support tools for tutors and educators is on high-level analysis for MOOC e-learning.

We have presented a design study on applying visual analytics to data from hands-on cybersecurity training in the form of CTF games. We introduced two iterations of the *Training Analysis Tool*, allowing tutors to assess the quality of the training scenarios and gain insight into the trainees' progress beyond the completion time and final score. The summative evaluation validated our design decisions. The verbal feedback from participants and the SEQ and SUS scores confirmed that the tools address the elicited requirements. We gradually learned more about what information tutors would like to display in the visualization and how they interact with the data during the design study. Based on this experience, we believe that a data-driven insight into the training courses could provide surprising insights and knowledge about the design and behavior of trainees.

Focusing on puzzle-games principles enabled us to conceptualize the data and visualizations beyond the cybersecurity domain. If we look closely at the information we used, we realize that it is a quadruple: timestamp, the ID of the trainee, type of event, content (arbitrary). Therefore, we believe that our approach can be easily applied in other areas where hands-on training becomes common. We admit that there are further requirements, such as automated processing of user inputs, but even basic logging can provide sufficient data. The level of detail depends mainly on the expressiveness of the content component.

Consider the university programming course as another application area. The tutors often evaluate students' assignments using automated compilation and validation tools against

predefined unit tests and datasets. The summary of code diffs, compiler error logs, and output of the automated tests can be logged. Similar to the cybersecurity domain, these events can be mapped to assessment events (e.g., penalties for unsuccessful unit tests), player actions (e.g., the submission of a piece of code), and progress events (e.g., successful compilation and test of a programming task). Visualizing these events on the timelines (one per student) or further text analysis of the code can be as valuable as our analogy with the cybersecurity CTF games.

The support tools for a category of so-called blended classrooms and hands-on courses are still mostly unexplored. Our work addresses only a tiny part of this broad research area. Despite our focus on cybersecurity education, we consider our findings applicable in other areas of puzzle-based learning and analyzing data from a single training session (i.e., low-level learning analysis). We want to encourage others to explore novel methods for visual analysis of puzzle-based learning courses in different areas.

The TAT is integrated into the user interface of the *KYPO CRP*. We also work on additional data integration from sandboxes (e.g., resource usage, executed commands, running processes). Enhancing the current level of event processing with this information will further improve the insight and enable a more detailed analysis of the training and its scenario. Our next goal is to explore the possibilities for visual analysis of multiple training sessions and analyze and assess trainees' long-term progress. Extending the analysis with automatic highlighting of anomalies or flaws in the training design is another direction of research that needs further study.

Acknowledgment

This work was supported by ERDF "CyberSecurity, Cyber-Crime and Critical Information Infrastructures Center of Excel-lence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822). Computationalresources were provided by the European Regional Development Fund Project CERIT Scientific Cloud (No.CZ.02.1.01/0.0/0.0/16_013/0001802).

Article G

Enhancing Situational Awareness for Tutors of Cybersecurity Capture the Flag Games

Karolína Dočkalová Burská[1], Vít Rusňák[2], Radek Ošlejšek[1]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2] Masaryk University, Institute of Computer Science, Brno, Czech Republic

iV – IEEE International Conference Information Visualisation. 2021, 8 pp.

Abstract

Supervised Capture the Flag games represent a popular method of practical hands-on training in cybersecurity education. However, as cybersecurity training sessions are process-oriented, tutors have only a limited insight into what trainees are doing and how they deal with the tasks. From their perspective, it is necessary to have situational awareness, enabling them to identify and react to any issues during a training session as soon as they emerge. We propose a tool designed in collaboration with cybersecurity educators. Based on user requirements, we developed the Progress Visualization Tool, which provides educators with timely feedback through the session. More specifically, the tool informs educators of the training progression, helps identify the students who might struggle with their tasks, and reveals overall deviation from the schedule. We validated the tool through formative and summative qualitative in-lab evaluations. The participants appraised the impact on the training workflow and gave further insights regarding the tool. We discuss the insights and recommendations that arose from the evaluations as they could aid the design of future tools for supporting educators, not only of CTFs but also in other domains.

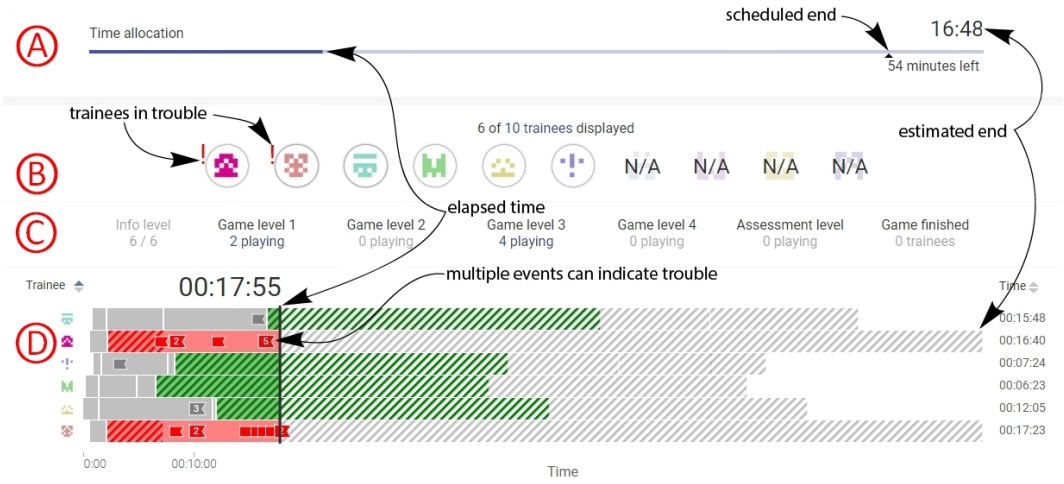

Figure G.1.1: Progress Visualization Tool (PVT) serves as a visual overview of ongoing hands-on training session. The tutors can quickly identify outstanding situations that may require their intervention. The tool consists of four sections which provide complementary information: A – Timeline, B – Trainees, C – Game Level Occupancy, and D – Detailed Timeline.

G.1 Introduction

Higher-order thinking has become one of the essential skills for the 21st century. The best way to develop and enhance these abilities is through practical hands-on courses [157, 155]. In the cybersecurity domain, hands-on learning is primarily represented by *Capture the Flag* (CTF) games [251, 59, 239]. Michalewicz et al. [162] introduced a game-based learning method that uses puzzles as a metaphor for getting students to think about how to frame and solve unstructured problems. In IT education, the puzzle-based learning approach is prevalent for many years [258, 159, 104]. Multiple studies confirm the usefulness of puzzle-based learning also for cybersecurity education [92, 107, 58].

Hands-on cybersecurity training is often organized in so-called cyber ranges [60, 249, 27]. The data and field observations referenced in this paper were obtained during the training sessions in *KYPO Cyber Range*[7] [263]. that we develop and operate since 2013. The Cyber Range is a cloud-based environment providing features for the virtualization of computer systems and networks. It serves as a platform for practical training of various cybersecurity skills, including regular CTF courses for students of our university.

CTF games can be organized in diverse ways. Popular are, for example, unsupervised online games where a trainee can access the game or interrupt it anytime. This paper, however, addresses blended CTF courses – tutored (or supervised) training sessions for small groups combining computer-supported learning activities with traditional face-to-face

[7]KYPO is a Czech acronym for Cybersecurity Polygon.

interaction.

Typical training is organized for 15–20 participants who individually solve cybersecurity tasks (puzzles), e.g., scan the network, identify a server, find the server vulnerability, exploit it, and gain the root privileges. A successful solution yields a short string (called *flag*). Entering the flag in the dedicated field opens the next puzzle. Struggling trainees can take hints specific for each puzzle or see the correct solution when helpless. Time for solving all the tasks is usually limited to the class length (one or two hours). Tutors walk around and help trainees either on request or when they realize that someone significantly lacks behind (typically by quick peek at their displays or asking them directly). In the end, the scoreboard displays individual scores, and the tutors hold a short debriefing with the presentation of correct solutions.

This kind of tutored CTF exercises become unexceptional in a formal cybersecurity education or professional training. However, a training session organization leads to cognitive and physical loads of tutors who overview the trainees' progress, need to recognize their difficulties and intervene in time. They also need to interact with trainees, make notes on their progress, and analyze them continuously in their heads. All of this makes teaching inefficient and error-prone.

Moreover, hands-on cybersecurity training is process-oriented. Other IT learning areas usually produce a tangible output that can be continuously checked, analyzed, and evaluated by the tutor, e.g., source code or results of unit tests in programming courses. On the contrary, during the CTF training sessions, tutors have only a limited view of what trainees are doing in the computer network and how they deal with the tasks. These circumstances make run-time supervision even more needed, but difficult.

In this paper, we present an interactive tool that captures and visualizes the progress of a CTF training session from only limited gameplay events in a way that helps tutors to gain insight into the training progress and manage the session efficiently.

G.2 Related Work

According to the classification provided by Oslejsek et al. [178], this paper addresses situation awareness of organizing participants (tutors). Using information technologies in blended cybersecurity courses enables us to collect the data that can be used by tutors for more targeted support during the training sessions. However, the design and deployment of efficient support tools remain a challenging problem [193]. Also, Macfadyen and Dawson [145] confirm the need for insight exceeding simple summative feedback to provide more focused and timely interventions. Our tool aims to fill the gap in providing real-time situational awareness for tutors of supervised CTF training sessions.

Govaerts et al. [94, 93] proposed a general-purpose web-based environment for the visu-

alization of Moodle activities to increase awareness and support self-reflection. Deeb and Hickey [62] utilize data of the web-based problem-solving learning environment to monitor students' performance in large classes. Their classroom orchestration tool allows tutors to monitor learners' progress on the given problem and visualizes equivalence classes and probabilities of transitions between incorrect attempts. However, these approaches address post-training feedback, which is important for situation awareness across multiple learning sessions. Our research focuses on efficient real-time support during a single training session when both students and tutors work under time pressure.

Holstein et al. [112] present a set of challenges for real-time teacher support systems. Despite the focus on K-12 math teachers, the challenges are valid in other areas as well. The challenges relevant for our scope address teachers' *needs to maintain control of their classrooms*, and their *desires to receive analytics informing them about their students' learning*. In their later work, Holstein et al. [111] addressed some of their challenges through an augmented reality system where teachers are wearing smart glasses that help them with personalized learning in classrooms.

A framework for real-time situation awareness based on interactive visualizations can be found in [153]. Their TrAVis system offers tools to monitor an individual or a group of students through the course and communication activities. The system is generic, supporting the whole analytical workflow and diverse data sources. The visual tools focus on many aspects, e.g., social, cognitive, and behavioral. On the contrary, our approach benefits from restricted application domain – a puzzle-based cybersecurity training, to provide a compact preview of the only aspects that may be significant for the educator's decisions at the moment.

Visual tools supporting the learning of low-level cybersecurity concepts can be found in the literature as well. These works focus on AES encryption and decryption [143] or access control models [246, 247], for instance. Their visual feedback helps the students to understand the taught concepts through a graphical interpretation, while the tutors can utilize them to assign exercises, quizzes, or to verify the students' results via a test report system. Our approach addresses any cybersecurity training content organized in the form of a CTF game.

The CyberPetri [14] is a prototype system for achieving situational awareness during cyber defense exercises. This work shares similar goals – providing real-time situation awareness for cybersecurity training. However, cyber defense exercises represent hands-on training based on group work, which is different from puzzle-based CTF games. Therefore, it cannot be directly used in the context of our work.

We address the lack of real-time support tools for tutors of the Capture the Flag games through the *Progress Visualization Tool* (PVT). The PVT enables real-time insight into students' behavior during the sessions and supports educators in managing the course progression and providing timely and focused guidance to the students.

G.3 Functional Requirements and Data Abstraction

We design the tool iteratively, guided by the design study methodology framework [211]. During the project, we closely collaborated with the cybersecurity educators from our university, who are also the target users. After initial interviews with three of them and field observations during the training sessions, we gathered the user requirements and analyzed the input data.

G.3.1 Functional Requirements

The interviews and field observations revealed that tutors would benefit from the better session timing foresight and seeing how the trainees perform. They require a glimpse of trainees' activities and performance to identify those who act unexpectedly or require assistance, without the need to disturb others. On the other hand, trainees' scores or their detailed assessment is unimportant at that moment. We formulated their needs during the training session on two primary functional requirements:

R1 – Training schedule overview: The tutors should overview the general situation of the training quickly. Especially time needed to finish the training (comparing it with the planned schedule) is important for the tutors to intervene in time. The tool should also provide a real-time overview of the training session, the expected duration of the training, the number of trainees in each level, and individual progress for all trainees.

R2 – Identification of at-risk trainees: Tutors should identify those who are behind the schedule or struggling with the puzzle at some level (e.g., entering multiple wrong flags, prolonged inactivity). The tool should display details of the actions performed by a trainee on-demand and enable the trainees' filtering based on their training duration and status.

G.3.2 Data Abstraction

We further identified two datasets used and generated during the training sessions that we can use as input sources: a training scenario and trainees' events.

The **training scenario** defines the content. It contains a background story, puzzle assignments (cybersecurity tasks), hints, hint penalties, solutions, solution penalties, correct flags, flag score points, and level time limits.

The **trainees' events** are automatically generated and stored when trainees play the game. Example events are: training started, training ended, level started, level ended, correct flag entered, the wrong flag entered, hint taken, solution taken. Each event contains a standard set of attributes (timestamp, event type, training description ID, training session ID, user ID). Three event types (a wrong flag entered, a hint used, a solution displayed) also contain specific attributes – a wrong flag string and penalty points, respectively.

G.4 Progress Visualization Tool Design

Based on the requirements analysis, we iteratively designed the tool. Further, we present its final design. The prototype, implemented using Angular and D3.js library, is available at `https://www.radek-oslejsek.cz/download/iV2021/` together with other supplemental materials.

The *Progress Visualization Tool (PVT)* is a single-page application organized into four horizontal sections (Figure G.1.1: timeline, trainees, game level occupancy, and detailed timeline). From top to bottom, each level adds more details to the upper ones.

G.4.1 Timeline

The timeline section (Figure G.1.1 – A) overviews a general timing in real-time (**R1**). The bold part (on the left) represents the elapsed time, while the arrow indicates the planned end of the training. Its position is updated regularly since the situation changes over time, and the training session might exceed the estimated schedule. The segment of the timeline right to the arrow denotes the estimated session overtime. The current estimated end time is displayed on the upper right side as a wall-clock time, while below there is the number of remaining minutes. Therefore, a tutor can quickly check how much time is left and detect the plan's deviation.

G.4.2 Trainees

The interactive list of trainees (Figure G.1.1 – B) helps tutors to see the status of all the participants (**R1**) and indicates those who need their attention quickly (**R2**). Tutors can display either trainees' names or avatars. Unique, auto-generated, immutable avatars provide visual identities alike profile pictures on community portals. The avatar is also displayed on trainees' user interface so the tutors, while walking around during the session, can easily connect avatars with them even if they do not know their names.

Until the trainees join the training session, their avatars are marked as "N/A". A circular outline marks the selected trainees whose details are displayed in the DETAILED TIMELINE section below. A red exclamation mark indicates that the trainee needs the tutor's attention. Currently, it notifies on three situations: being behind schedule for the current level by more than half of the estimated level duration, taking all level hints, submitting five or more wrong flags. A tooltip shows which situation(s) occurred on mouseover. New notifications for other states can be implemented if needed (e.g., a long period of inactivity without taking any hints).

G.4.3 Game Level Occupancy

Arranged in a horizontal list, the game level occupancy section (Figure G.1.1 – C) provides another degree of awareness for **R1**. It helps tutors to indicate possible latecomers based on the presence of trainees in levels. By clicking the level occupancy, the tutor filters out all the trainees but those currently playing the level from the lower DETAILED TIMELINE. A mouseover pop-up displays the level name and respective correct flag.

G.4.4 Detailed Timeline

The last section provides a detailed view of the trainees' progressions and activities in a compact and uncluttered way. The detailed timeline (Figure G.1.1 – D) resembles stacked bar charts where each row corresponds to one trainee's data. Segments represent training levels and encompass related game events. A black vertical line indicates the elapsed time.

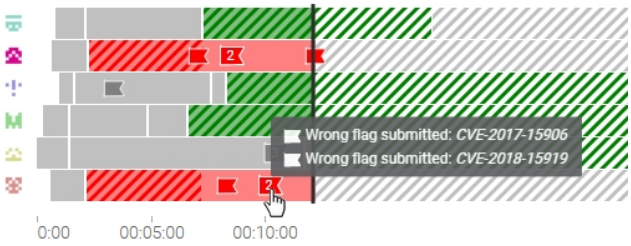

Figure G.4.1: A detailed timeline with aggregated events and details shown on demand.

As the training advances, the bars grow and display the trainees' current state and activities. The striped segments represent the scheduled time frame of the ongoing and following levels to promote **R1** from another perspective. Moreover, each trainee's current level has a specific color related to the fulfillment of the level's schedule. The color changes from green to orange when being over an estimated level time and to red when exceeding the estimate 1.5 times. Once finished, the level section becomes gray. This behavior highlights only relevant information and identifies the trainees who struggle with the current task (**R2**).

To indicate the training events related to individual levels, we use glyphs inside the segments (Figure G.4.1) indicating three situations: submission of wrong flags (displayed as a flag), taken hints (circles), and displayed solutions (checkmarks). A mouseover pop-ups a tooltip with additional information.

Events of the same type occurring in a short time can indicate trainees in trouble (**R2**). A typical situation is when a trainee attempts to guess the flag continuously. The visualization aggregates these events showing their count inside. The aggregation helps to unclutter the timeline and emphasize areas to which the tutor should pay attention. Long spans of trainee's inactivity can also indicate trouble in solving the puzzles. The tutor can always zoom in to

expand the timeline visually.

G.5 Design Decisions

When designing the PVT, we put emphasis on using the situation awareness design principles specified by Endsley [79]. We primarily draw attention to eight principles that are relevant to the purpose of the application and features of available data.

Organize information around goals: The PVT consists of four mutually connected segments. Segment (A) provides tutors with an overview of the overall training schedule, while segments (B)–(D) primarily allow tutors to identify at-risk trainees at various levels of detail and from different perspectives.

Present derived information directly to support comprehension: Many derived pieces of information that are key for decision making of tutors are provided directly, e.g., trainees in troubles are explicitly highlighted, the remaining time of the session is estimated and updated regularly.

Provide assistance for data projections: Features like colors of levels changed dynamically with respect to the schedule of the training help tutors to project future development of the training session (multiple red levels, for instance, can indicate trouble in complying with the time reserved for the training session).

Support global situation awareness: The PVT is a compact application providing a complete overview of the situation on a single standard FullHD screen. No pop-ups or multiple windows are used.

Support trade-offs between goal-driven and data-driven processing. Initially, tutors see a global overview of the situation. Exclamation marks indicate situations worth investigating and thus provide attentional narrowing (top-down processing). However, a tutor can decide to process the situation bottom-up. Detailed information of all trainees can be displayed to let the tutor choose a new investigation goal according to their specific walkthroughs.

Make critical cues for schema activation salient: Two critical cues, i.e., trainees in trouble and the delay compared to schedule, are explicitly indicated and highlighted in the tool. These cues usually force tutors to act, either help a particular trainee or give general hints or explanation to the whole study group.

Use information filtering carefully: The filtering rules have been chosen with respect to the importance of the information for runtime decisions making in training programs. For example, values of submitted flags are hidden, remaining available as tool-tips on demand.

Explicitly identify missing information: Trainees who did not start yet (their data are not available) are displayed to tutors so that they can identify missing participants or users with technical difficulties. The tutors can also spot trainees' inactivity from the DETAILED

timeline section.

G.6 Evaluation

We conducted two qualitative user studies. We created the early prototype and performed a qualitative formative evaluation with five collaborating cybersecurity educators and one student familiar with the CTF games. Our goal was to assess the usability and usefulness of the visualization, gather feedback on how the tool fulfills the two requirements, and identify possible refinements for the next design process iteration.

We then added new features and redesigned the user interface of the tool based on received feedback. A qualitative summative evaluation with eight participants served us for the validation of the final design.

G.6.1 Participants

The target users of the PVT are domain experts with necessary background knowledge (e.g., terminology, game design). We thus recruited five cybersecurity educators and three students who passed the CTF design course taught at our university. The educators also organize university courses, training events for practitioners, or both. The students represent novice users familiar with the cybersecurity CTF games and their basic concepts. They also have hands-on experience with their design. Note that P1–P3 participated during the requirements analysis stage, and P5 co-authored the training scenario of the dataset we used during the summative evaluation. Also, P7 and P8 participated only in the summative evaluation. The average age of the participants was 27.6 years (*SD=4.1*), and the average teaching experience was 4.8 years (P1–P5 only).

G.6.2 Procedure

The procedure was the same for both formative and summative evaluation. We held the formative evaluation sessions in person. The experimenter took notes and audio recorded the participants' opinions and thoughts. The summative evaluation was, due to the pandemic situation, conducted online using Google Meet, which we also used to record audio and screen. The sessions lasted 40–60 minutes and had three parts.

In the introductory part, the experimenter explained the evaluation procedure, and the participant consented and filled the demography questionnaire. The experimenter then presented the tool, and the participant spent 2–3 minutes familiarizing with it using dummy data.

Next, the experimenter introduced the two tasks addressing requirements **R1** and **R2**:

- *T1: Identify trainees in trouble, make an assumption of their cause, and conceive your reaction.*
- *T2: Identify problems that can influence the overall training session duration. What is their cause, and what would be your reaction?*

During the main part, the participant was asked to think aloud and comment on the current situation and suggest the (re)actions. We used the real datasets and integrated a re-play feature to visualize the trainees' activity dynamically. We also sped-up the re-play timing ten times to reduce the study session's overall length and mimic the situations when the tutor does not pay full attention to the tool. Even so, the participants were able to follow the situation without any problems.

Finally, the participant filled the usability questionnaires and debriefed on final thoughts and feature requests. We chose the SUS – System Usability Scale [202] and the SEQ – Simple Ease Question [203], two widely used questionnaires for measuring various products' usability. The former is a de facto standard method for assessing the usability of various tools or systems. The latter helps to quantify the usability of individual tasks.The SEQ is also considered as a powerful measure when the number of participants is low and for tasks that are too complex for metrics like task duration time or completion rate [203].

G.6.3 Datasets

We used three datasets from real training sessions in the main part. DS1 and DS2 were used in the formative evaluation, DS3 in the summative one. All the datasets contain various actions observable during training sessions (e.g., guessing the correct flag, prolonged inactivity, varying performance of trainees).

DS1 was from the tutorial on computer forensic skills and consists of six game levels. The goal is to identify and examine malicious software running in the computer system. The trainees learn how to identify a suspicious application, dissect its executable, and process memory. The session had 16 trainees and lasted 55 minutes. It generated 374 events. DS2 was an attack-oriented training scenario that consists of four game levels with the following puzzles: exploit server vulnerability, gain the root privileges, access a protected data file, and cover the traces after the attack. Six trainees participated in this session and generated 146 events. This training took 90 minutes. DS3 uses data from a training session held as the introductory lecture of the CTF game design course. It is an attack-oriented four-level training scenario analogous to DS2. In this case, nine trainees generated 281 events during the session lasting 110 minutes.

We provide DS3 in the supplemental material. The dataset consists of the anonymized[8] training scenario data (description of the tasks, scoring, etc.) and events generated by

[8] We replaced hints and solutions with dummy texts and modified correct flags.

trainees as described in Section G.3.2. DS1 and DS2 cannot be published due to content protection policies.

G.6.4 Results

The participants performed without struggles. Their immediate feedback was very positive. Further, we present the evaluation outcomes, and findings resulted from an inductive qualitative analysis [226] of the recordings.

PVT is easy to learn and offers a great user experience. The SUS score increased from 79.2 in the formative evaluation to 87.8 in the final summative evaluation (i.e., an *excellent* rating according to the adjective ratings [18]). Moreover, low scores of the questions "I think that I would need the support of a technical person to be able to use this product" and "I needed to learn many things before I could get going with this product" can be interpreted as good learnability [202]. The SEQ score medians were 6.5 (*T1*) and 5.5 (*T2*) in both evaluations, suggesting that the PVT provides good support for the two typical tutors' tasks.

PVT streamlines the workflow and reduces the time needed to gain situational awareness. All the participants were checking the notifications frequently, as they *"immediately indicate that something is going on"* (P5). An additional look on the DETAILED TIMELINE gave them further context necessary for the suggested action. We also observed extensive use of level filters providing necessary selection and enable comparison of players at the same level. The participants either went through the levels to quickly overview whether someone is overdue or focused only on the slowest trainees. They usually continued with the detailed inspection of trainees in DETAILED TIMELINE.

PVT provides an early indication of the potential delay. The participants were well-informed on the current training session delay even though they checked the timeline (Figure G.1.1 – A) spontaneously. However, we noticed that the main trigger for intentional time control was trainees overdue indicated by orange/red color in the DETAILED TIMELINE. P1 expressed that *"[it] is the main feature that helps prevent training session delay."* Whenever participants found out that one or more trainees are overdue with the current level, their typical reaction was that those trainees should immediately take some hints (when orange) or solutions (when red). Moreover, the growing portion of displayed orange (or red) color also increased the urgency for a reaction. We also noted that when more trainees were delayed at the same level, some participants (P2—P4) tried to figure out if there is some common problem or several unrelated ones.

PVT supports the decision-making process. Tutors tend to focus on the slowest trainees since they cause the training delay frequently. The presence or absence and distribution of glyphs on the timeline provide necessary input for the decision process leading to more focused advice. For example, P3 remarked *"I clearly see that these trainees don't*

take hints and are running late, so I would advise them to do so immediately ... and here is a bit of frustration since the player took all the hints at the very beginning in the last two levels"). P4 advocated the aggregation of the same events by saying *"the aggregation of multiple flags is also good; it shows me whether the trainee tries to guess the flag or struggles with the correct format of the string."*

Gaps and drawbacks. We observed no strong preference for neither the avatar nor the textual representation of trainees. P3 remarked that *"the avatars are useful"* while P1 and P7 would appreciate displaying avatar with the name/ID. The participants also suggested minor improvements such as adding the markers for the expected duration of each level to the timeline (P5) or enable *"to mark notifications as read"* (P1, P3). The green-orange-red coloring highlights only the current level. Especially in the late phase of the training session, multiple trainees were delayed but in different game levels. P1 and P3 remarked that *"it is uneasy to identify in which level trainees are."* Nevertheless, the participants used the level filters to overcome the issue without hesitation.

G.7 Discussion

Without the PVT, tutors maintained situational awareness in their heads. They were dependent on time-consuming and inefficient written notes and physical observations (literally by "looking over trainees' shoulders"). Advice to individuals was rare and usually only on trainees' requests since they mostly advised the whole group. Our approach reduces tutors' cognitive and physical demands and provides them timely insight into the training session.

Further, we present the study limitations and propose implications for designing similar tools. We also discuss how such tools can be generalized to related IT courses.

G.7.1 Study Limitations

Both user studies had two main limitations to the external validity: the low number of participants and the simulated execution of the training sessions instead of the ex-situ field evaluation.

To ensure the evaluation's ecological validity, we needed users with practical experience with organizing hands-on training sessions and knowledge of cybersecurity education. These demands notably restrict our choice of suitable candidates. Our collaborating cybersecurity educators are, no doubt, the primary users of the developed tools. Therefore, they provided relevant feedback, which will serve as a source for our further thoughts on both tools' improvements.

Hands-on training events are not organized frequently at a scale suitable for proper field evaluation, especially during the last year due to the pandemic situation. Therefore, we

decided to realize the in-lab studies using real-world datasets to emulate the real conditions instead.

G.7.2 Generalization to Related Courses

A big effort has been made in the past to conceptualize data mining and digital assessment for serious games so that generic learning analytics principles can be researched and applied regardless of the specific game content [52, 12, 180]. Our solution deals with event logs and the score-based assessment that represent broadly used types of telemetry and evaluation data for serious games.

If we look closely at the information we used, it is a quadruple: timestamp, the ID of the trainee, type of event, content (arbitrary). Even basic logging can provide sufficient data, and the level of detail depends mostly on the expressiveness of the content component.

Consider the university programming course as another application area, for instance. The tutors often streamline the tasks' evaluation via automated compilation and validation against predefined unit tests and datasets. What can be logged are: summary of code diffs, compiler error logs, and output of the automated tests. Visualizing these events on the timelines (one per each student) or doing further text analysis of the code can be as valuable as our analogy with the cybersecurity CTF games.

Therefore, we believe that our approach can also be applied in other areas where hands-on training becomes a common practice.

G.7.3 Design Implications

During the project, we gradually learned more about what kind of information tutors would like to display and how they want to interact with them. In addition to the identification of typical tasks and user requirements, we elicited three design implications for similar tutor supporting tools:

- *Intuitiveness over complexity.* The tool should be intuitive and easy to use, not to divert tutors' attention from the class. The tutors' main goal is to guide the trainees, interact with them, and intervene if necessary.
- *Notifications.* Identification and highlighting of notable events (e.g., exceeded estimated level duration, too many wrong attempts) were among the most appreciated features in PVT. Notifications are a convenient method to attract tutors' attention.
- *Sorting and filtering.* Based on real usage scenarios, the tool should provide sorting and filtering options so that tutors can quickly focus on a particular issue.

G.8 Conclusion and Future Work

The support tools for tutors' assistance during a training session are mostly unexplored. Our work addresses only a small part of this broad research area. We introduced the *Progress Visualization Tool* that improves the tutors' insight during the hands-on cybersecurity training sessions and helps them in more targeted feedback to individuals. The verbal feedback from user study participants and the results of usability questionnaires validated our design decisions and confirmed that the tool addresses the elicited requirements.

The PVT has been designed for on-site training. However, the tool has been used successfully also for the training sessions held remotely due to the COVID-19 pandemic. It would be virtually impossible to organize supervised CTF sessions online without the runtime insight into the trainees' actions provided by the PVT.

Article H

Exploratory Analysis of File System Metadata for Rapid Investigation of Security Incidents

Michal Beran[1], František Hrdina[1], Daniel Kouřil[1], Radek Ošlejšek[2], Kristína Zákopčanová[2]

[1] Masaryk University, Institute of Computer Science, Brno, Czech Republic
[2] Masaryk University, Faculty of Informatics, Brno, Czech Republic

VizSec – IEEE Symposium on Visualization for Cyber Security. 2020, 10 pp.

Abstract

Investigating cybersecurity incidents requires in-depth knowledge from the analyst. Moreover, the whole process is demanding due to the vast data volumes that need to be analyzed. While various techniques exist nowadays to help with particular tasks of the analysis, the process as a whole still requires a lot of manual activities and expert skills. We propose an approach that allows the analysis of disk snapshots more efficiently and with lower demands on expert knowledge. Following a user-centered design methodology, we implemented an analytical tool to guide analysts during security incident investigations. The viability of the solution was validated by an evaluation conducted with members of different security teams.

H.1 Introduction

Cybercrime has rapidly developed over the past years [89], and cybersecurity threats are expected to present significant risks for the future [13]. For computer systems to be able to face the constantly changing threat landscape, it is necessary to develop and maintain

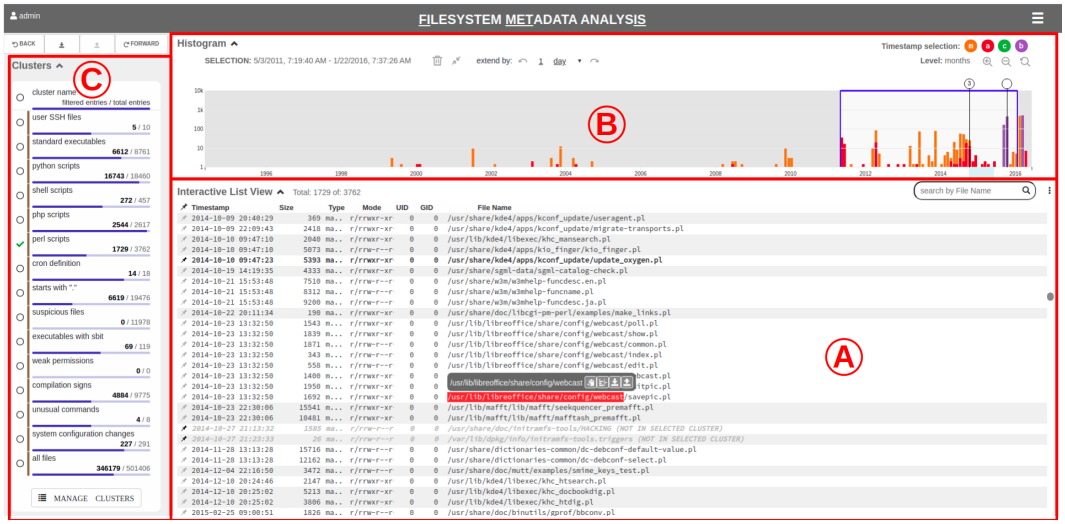

Figure H.0.1: FIMETIS is a tool providing an interactive exploration of file system snapshots. Analysts can quickly investigate cybersecurity incidents via three complementary views: A – *list view* with file system records, B – *histogram* with a timeline, and C – data *clusters*.

capabilities for responding to cybersecurity attacks. A vital part of the response process consists of the investigation of the evidence, which reveals the nature of the incident and performed activities.

The investigation depends heavily on a proper evaluation of all collected evidence. Methods of digital forensics [44, 122] are employed for systematic scrutiny of the data. It is a continuous process where hypotheses are formulated based on observations followed by steps to either confirm or deny the theory.

A simplified scheme of an investigation workflow is depicted in Figure H.1.1. First, the suspicion of an incident is reported in the form of a preliminary report. Then, data sources for digital evidence of the incident are collected. They capture either the broader state of involved computer networks and communication history (net flows, PCAPs) or the state of involved devices (system logs, the content of disks, memory snapshots, etc.).

The iterative investigation is often time-consuming and requires a high level of expert knowledge. The amount of data collected is often high, which only complicates the analysis. While the forensic investigation methods provide a great platform to derive particular results, a user-oriented approach is missing to simplify the overall process.

Permanent storage devices are a crucial part of contemporary computer systems and data retrieved from these devices provide significant input for the investigation. The state of permanent storage can be captured in multiple ways. The most straightforward and complete approach is to analyze the complete disk content. However, as current media tend to be

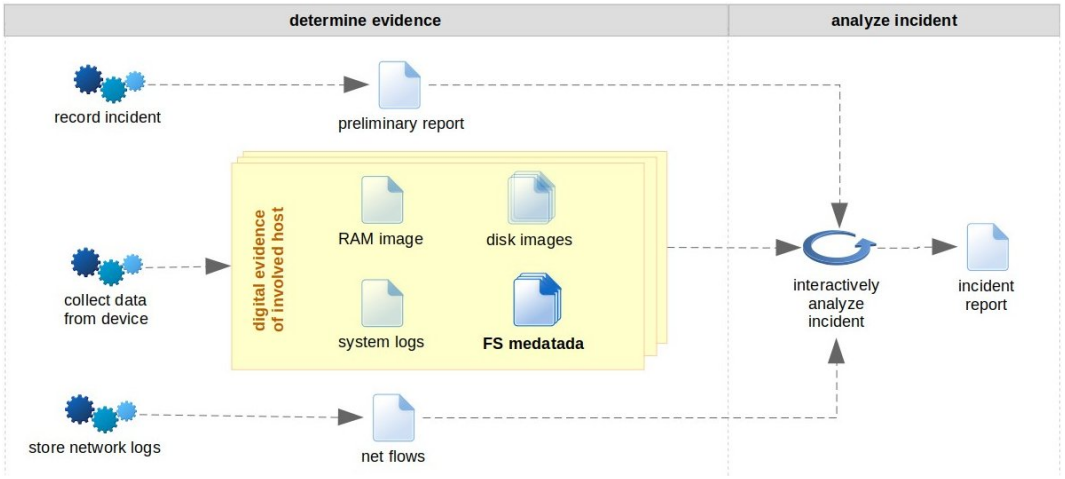

Figure H.1.1: Incident investigation process. The *FIMETIS* tool deals with file system metadata only.

quite large—it is not uncommon for disks to provide several terabytes of capacity—the analysis becomes time- and resource-demanding. Moreover, analyzing disk content encounters privacy issues when the data contain sensitive information [45].

One way of coping with the volume and privacy problems is to work only with file metadata, extracted from permanent storage, which include the file owner, size, name and dates of last manipulations. However, even though such a dataset is much smaller in size compared to raw disk images, it is still necessary to process hundreds of thousands of records already in case of a standard storage. Moreover, it requires deep knowledge about the relationships among files, their purpose in the system, and importance for the attacker.

In this paper, we propose visual-analytic methods that make the investigation of file system metadata significantly more efficient and are also available to analysts with no deep domain knowledge. We describe an application called *FIMETIS* (FIlesystem METadata analysIS) that was developed to verify the visual-analytic concepts. Evaluation of this tool has shown that the user interface is easy to learn and well supports analytical tasks. Even less skilled participants were able to investigate and reconstruct a real incident in limited time at surprising precision and level of details.

H.2 Related Work

Many tools and approaches dealing with individual types of data sources for digital evidence can be found.

So far, big attention has been paid to the investigation of network communication. NetCapVis [229] provides a post-incident visual analysis of PCAP files that capture network

traffic. TVi [34] is a tool that combines multiple visual representations of network traces to support different levels of visual-based querying and reasoning required for making sense of complex traffic data. Visualization techniques proposed by Gray et al. [97] provide conceptual network navigation for situational awareness in network communication.

Analysis of system logs was researched as part of ELVIS [114] and CORGI [115], for instance. These tools, both proposed by the same authors, provide security-oriented log visualizations that allow security experts to visually explore and link numerous types of log files through relevant representations and global filtering. A top-down approach to the log exploration is provided by the Visual Filter [221] tool, which represents the whole log in a single overview and then allows the investigators to navigate and make context-preserving sub-selections.

Disks and permanent storage provide another valuable source of information for the digital investigation. Disk and file systems analysis can be performed in several layers [43]. Approaches addressing specific features are, for example, Change-link 2.0 [139], which provides several visualizations to capture changes to files and directories over time, or the work of Heitzmann et al. [106], who proposed a visual representation of access control permissions in a standard hierarchical file system using treemaps.

This paper deals with the utilization of file system metadata as they have lesser demands on volumes and do not threaten data sensitivity. The utility of metadata for digital forensics has been articulated previously [37], and various techniques for metadata-based analyses have been proposed since then. The use of metadata to provide a fingerprint of actions performed with files has been suggested to streamline file system analysis [121].

Metadata attributes are also known to be useful to reconstruct a timeline of previous activities [103] and have been demonstrated to locate suspicious files [195]. These techniques address the particular sub-problems of the analysis. To facilitate the whole investigation process, it is necessary to support interactive work, which would support the above-mentioned analytical techniques and make them easily accessible to users.

Only a few papers can be found on approaches supporting interactive work with the data of digital evidence, which is essential for the whole forensic investigation process. Our literature survey revealed two works dealing with timelines constructed from file system activities, which are very relevant to our research.

The Zeitline [38] tool represents activities as generic events. The user interface enables analysts to group events and then make the timeline hierarchical, to filter obtained data trees, and locate specific events by queries.

In the CyberForensic TimeLab [175], the timeline is implemented as a histogram using bars to represent the number of pieces of evidence at a specific time. The investigator can highlight interesting parts of the timeline and zoom in to get greater detail of that particular time span.

Both the tools are designed as generic, enabling analysts to create timelines from multiple resources, e.g., from file system metadata as well as system logs, and their user interfaces reflect this universality. In contrast, our approach focuses solely on file snapshots build from metadata only. We aim to make the analysis of this specific data maximally effective, focusing not only on the timeline but also on other data available for files. To reach this goal, we follow a user-centered design methodology, which is extended with a mechanism guiding the investigator during the process. Although our design shares some visual elements with the CyberForensic TimeLab, e.g., histograms, our solution provides an interface fine-tuned for a single specific use case – a forensic analysis of file system snapshots. On the other hand, the visual-analytics concepts proposed in this paper are sufficiently general that they could be extended to other types of timeline in the future.

H.3 Design Methodology

In this project, we applied the user-centered approach guided by the design study methodology framework [211], mainly reflecting its *core* stages: discover, design, implement, deploy.

In the *discover* stage, we gained a better understanding of the workflows of the digital investigation and elicited user requirements on the tool in order to simplify the analytical tasks.

The initial insight into the application domain was provided by a co-author of this paper, who is a member of the cybersecurity team of Masaryk University. Based on his initial input, we conducted semi-structured informal interviews with two other domain experts who have long-term experience with practical investigations of cybersecurity incidents. The first respondent works as a senior security specialist at CESNET – an academic institution in the Czech Republic providing IT services to Czech academia. The second expert is a member of the incident response team at Masaryk University. All three of them have long-term experience with practical investigation of cybersecurity incidents. Each interview lasted about two hours.

Based on these interviews, we distilled a generic workflow of the investigation process and formulated requirements for a file system analysis. The results are presented in Section H.4.

In the *design* stage, we proposed the visual elements and the interactive dashboard reflecting the functional requirements. The design was proposed and refined iteratively. User interfaces were continuously prototyped under consultation with the domain expert (co-author of the paper). Proposed visual encoding is described in Section H.5.

In the *implement* stage, we iteratively developed the analytical dashboard. We paid attention to the observation that cybersecurity experts investigate incidents rarely, and evidence collection is a long-term interactive process. Architecture and implementation of the tool are described in Section H.6.

In the *deploy* stage, we evaluated the tool. As the investigation of real cybersecurity incidents is a sensitive process, we could not perform a usability study *in the wild*. Moreover, as the developed tool deals with only part of this process, we conducted a qualitative evaluation focused directly on the tool. However, we used data from a real incident. The evaluation is described in Section H.7 and results are summarized in Section H.8.

H.4 Requirement Analysis

The interviews conducted during the *discover* stage of the design methodology revealed that incident investigators would benefit from an interactive tool for file system exploration. Specific requirements were inferred from the characteristics of the data and the analytical workflow.

H.4.1 Data Characteristics and Abstraction

The investigation of cybersecurity incidents aims to provide answers to key questions related to the incident, like when the activities happened, what data was changed during the incident, where the activities originated from, etc. The process of investigation is driven by methodologies stipulated by digital forensics. The whole process comprises three main stages during which the evidence is acquired, analyzed, and the final report is produced. A simplified schema of the process is depicted in Figure H.1.1.

During the acquisition phase, the investigator needs to identify and collect the data that is likely to provide evidence about the case. The number of possible data sources from which digital evidence can be collected is vast. In case of forensic examinations performed directly on the machine, it is common to gather data from permanent storage (hard disk or external device like USB storage). There are also other sources of digital evidence, such as network traffic or its metadata, state and content of volatile memory, or information about authentication attempts. The rest of the paper deals with analysis of files and their metadata. It keeps the investigation domain limited in size while making it possible to evaluate the main principles.

File metadata describes information about the file, maintained by the operating system together with the file data. The exact scope of metadata depends on the operating system used, however, nowadays, it is common for all widely used file systems to recognize the file name, file ownership (specifying the user and a group), content size, and access rights. Besides these, several timestamps are maintained, indicating the time when key activities with the file or the metadata were last performed:

- *a-time*: the time when the file content was last read (accessed),
- *m-time*: the time when the file content was last modified,

- *c-time*: the time when the metadata record was last changed (e.g., during the change of access rights),
- *b-time*: the time when the file was created. The *b-time* timestamp is supported only by advanced file systems.

All the timestamps, except for *b-time*, change during the file life-time based on the operations performed. When a timestamp is updated, the previous value is overwritten and lost, which means they always refer only to the last performed actions.

Timestamps are an essential source of information for the reconstruction of events relevant to the investigation. They can help understand when certain operations took place but also reveal the nature of the activities performed. For instance, when a file is copied from another computer, the copying process usually retains the original timestamp. Such a file has the *m-time* value set to a date before the *b-time* and *c-time* values, which both will refer to the time when the copying process finished. A brand-new file created on the system has all the timestamps set to the same value upon creation. The difference in the timestamps can reveal where the file originates from.

Even if they do not reveal the actual file content, all file metadata attributes play a big role in the incident analysis. One of the most important reconstructions is determination of the timeline of actions performed in the analyzed system. A timeline emphasizes crucial activities conducted during the incident. For instance, it specifies when the attacker accessed the system for the first time or when a specific system configuration got changed.

A timeline constructed from metadata is a list of records ordered by the timestamps. Since there are multiple timestamp types assigned to a file, a single file can occur multiple times in the list, whenever its timestamps differ. A typical timeline contains hundreds of thousands of records, which need to be further analyzed.

In addition to providing input to recover the timeline, metadata can be used for efficient filtering of files, based on unique *fingerprints* they form, such as similarities of file locations, common access rights, or suspicious ownership.

H.4.2 Requirements

Based on the interviews, data abstraction, and the analytical workflow, we identified five functional requirements:

R1: Exploration of the file system structure. During the investigation, the analysts have to pay attention to different parts of the file system, e.g., files in a specific directory, files with specific extensions, or all log files. However, the interviewed domain experts emphasized that the interactive hierarchical exploration of the file system is not helpful. Instead, they need a global temporal view of the file system data with the possibility to navigate in the file system structure effectively. The analytical tool should support analysts

in the efficient switching between different parts of the file system and narrowing the area of interest by offering filtering functions that would localize the data by various aspects and meaning encoded in the available file system metadata.

R2: Exploration of temporal relationships. Disk snapshots have strong temporal characteristics. Each record provides the timestamp of the last manipulation, e.g., the creation, modification, or access. However, every file or directory usually appears multiple times in the dataset as the manipulation timestamps differ, which increases the data volume to be inspected. Also, the recorded data period is often very long, containing timestamps from a time long before the system was installed (but from when the files were created). Therefore, providing a scalable temporal view on the data with efficient filtering, zooming, and preserving time coherence is very important for making the analysis effective.

R3: Detection of file system anomalies. Some combinations of file locations and attributes can be considered unusual or deserving analyst's attention. For example, publicly writable files or directories, hidden files outside of users' homes, executables with administrator's privileges, files masking their names (e.g., a binary file with a *.txt* extension or named with only white spaces). The analytical tool should provide multiple views on various combinations of location paths and attributes in order to localize potential anomalies easily, and then further explore the corresponding files using **R1** and **R2** principles.

R4: Traces of the execution of suspicious commands. Some commands are seldom used by administrators but often used by attackers. For example, the `shred` Unix command is often used to wipe data content. The tool should allow analysts to verify whether or not such commands were used. Command execution can be identified by the *a-time* attribute. Once the command execution is confirmed, the analyst can use interactions reflecting **R1** and **R2** to explore details, analyze the impact of the execution, and either confirm or reject the hypothesis that an attacker executed the command.

R5: Traces of batch processing. Besides the execution of specific commands (**R4**), attackers often use scripts to perform reconnaissance on the system or to compile programs or libraries before installing them into the system. These batch activities can be recognized by the execution of multiple commands or the creation of multiple files in a short time, while manual tasks take a longer time. However, batch processing can represent a legal activity, e.g., the legal compilation or the result of regular system updates. Therefore, the tool should support analysts in efficiently identifying batch processes in the huge amount of file system data and then allowing them to analyze suspicious activities further using **R1** and **R2**.

While the requirements **R1** and **R2** reflect the generic investigation workflow, requirements **R3**–**R5** are related to more specific analytical questions that are often asked during the file system investigation. Besides these functional requirements, we set two complementary qualitative requirements that affect the architecture and implementation. These requirements follow the practice emphasized by the interviewees where cybersecurity experts investigate incidents rarely, and every investigation takes a lot of time (hours or days).

R6: Easy to use. Even practicing incident investigators analyze disks rarely (see Section H.7). Therefore, they should be able to use the tool even after a long period without the need for repeated learning.

R7: Persistence. The data and interactions have to be persistent so that an analyst can pause the investigation process and continue later on. Persistence is also important for recalling previous investigations and comparing hypotheses and results.

H.5 Visual Design

In this section, we summarize the design rationale, visual encoding, and interaction capabilities. The user interface consists of three coordinated views [192, 205], where a change in one view to the dataset affects other parts of the dashboard.

H.5.1 List View

The *List View* (Figure H.0.1 – A) is a dominant part of the dashboard providing a view on the raw data. Records are sorted by the timestamp by default (**R2**), but they can be re-ordered according to the file system structure (**R1**) by clicking on the *File Name* or *Type* columns. Individual columns can be shown or hidden via the *List View* menu (the three dots in the up-right corner of the *list view* area).

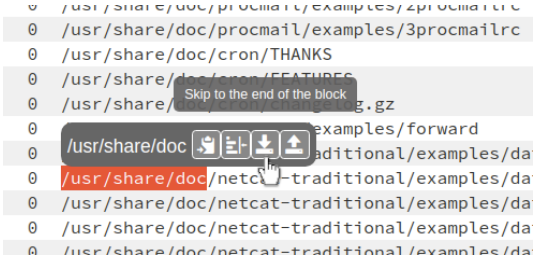

Figure H.5.1: Detail of smart block skipping in the *List View*.

Analysts can browse records traditionally by scrolling the list up and down, or they can use *smart block skipping* (Figure H.5.1) that significantly increases the efficiency of the list exploration. By clicking on a timestamp or a file path, the prefix is highlighted, and a context menu appears that enables analysts to skip records with the same prefix. Using this feature, analysts can quickly navigate to the next or previous date, hour, or sub-directory, and then accelerate the data exploration either from structural (**R1**) or temporal (**R2**) perspective.

The background of lines with the same timestamp is brushed to visually distinguish different time blocks (**R2**).

Search operation in the list works at two levels (the *name selection* label in Figure H.5.2).

Typing text into the input search field highlights the corresponding parts of the file paths. If the text is confirmed or the user clicks at the magnifier icon, then the list of records is filtered out, and only relevant lines remain displayed, enabling the analyst to pay attention to only desired files and directories (**R1**,**R4**). Data filtered out in this way remains in the *Histogram* (see subsection H.5.2) to preserve a broader context, but they are grayed out.

Records of high importance can be bookmarked (the *bookmarks* label in Figure H.5.2). Bookmarked records are emphasized in the list, displayed in the *Histogram* view, and used for fast navigation (**R2**). Bookmarks are persistent throughout the whole analysis and can be removed only on demand. Moreover, as they provide a broader context with significant events, the bookmarked lines are always visible in the *List View*, even if they do not fit all filters of the dashboard at the moment.

H.5.2 Histogram

The *Histogram* section (Figure H.0.1 – B) provides an interactive view on data distribution.

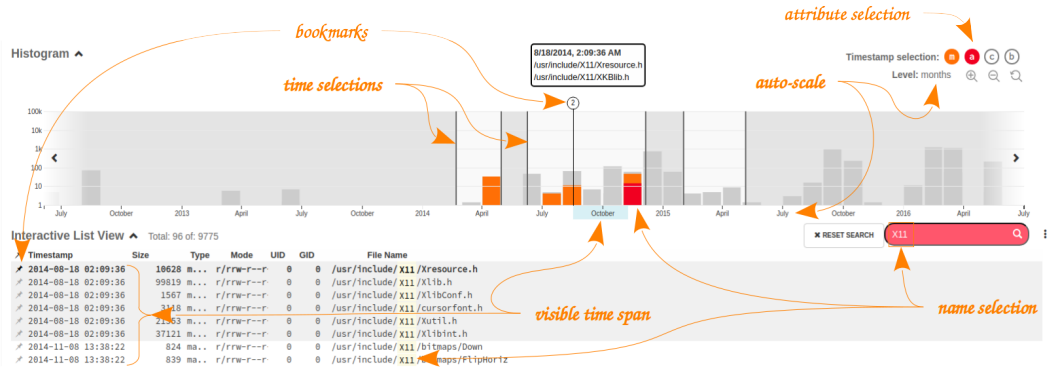

Figure H.5.2: Navigation and filtering in the *List View* and *Histogram*.

The y-axis encodes the number of records. The axis has a logarithmic scale to deal with high peaks that often appear in the data but still preserve the visibility of low numbers that can be important for analysts.

The x-axis is scaled automatically (the *auto-scale* label in Figure H.5.2). When zooming in, the x-axis automatically changes from years to months, days, and hours, and vice versa. The bars are recalculated and aggregated accordingly, representing the distribution in a specific year, month, day, etc. Zooming can be performed either by mouse, keyboard, or via icons in the upper-right corner.

Different colors in the histogram encode different file system operations (values of the *Type* column in the *List View*). Color encoding is shown in the *Timestamp selection* section. A detailed description of the metadata attributes is provided when the mouse is located over an icon. Similarly, hovering the mouse pointer above a bar in the histogram triggers a pop-

up tool-tip with attribute type, time, and an exact number of records. Clicking on a bar scrolls the *List View* to the corresponding entries.

The *Timestamp selection* is also used for per-attribute filtering (the *attribute selection* label in Figure H.5.2). Attributes can be switched on or off in the histogram by clicking on the icons. The *List View* is updated accordingly – only the records with selected attributes are shown in the list.

The histogram also serves as a time focusing tool (the *time selection* label in Figure H.5.2). Using a mouse, the analyst can draw multiple span windows and thus restrict the lines shown in the *List View*. A context menu appears when a user selects a selection span window. This menu enables the user to perform common operations, like extending the span, zooming into the span, or erasing the span. Some of these operations are available via direct mouse interaction in the histogram as well.

Due to restricted space on the web page, the *List View* displays only part of all the records at any one time (the rest is available via scrolling). Visible records represent span, which is emphasized in the x-axis of the histogram as a cyan stripe (the *visible time span* label in Figure H.5.2). This stripe supports the visual correlation between the *List View* and the histogram.

Entries bookmarked in the *List View* are shown in the histogram as push-pin icons. If they are too dense, they are aggregated into a single icon with a number of merged bookmarks. Details are provided as a tool-tip triggered on the mouse hover. Click on the icon scrolls the *List View* into the corresponding entry (to the first record in the case of aggregated push-pin). Push-pins that are out of selection spans are not clickable.

Span selectors, bookmarks, and automatically adaptable x-axis represent a powerful combination enabling analysts to scale and explore data from the time perspective (**R2**).

The structural exploration (**R1**) is less dominant in the histogram view. It is mainly restricted to the per-attribute filtering of records. On the other hand, the per-attribute filtering combined with the path filtering of the *List View* provides a generic approach to solve **R3** and **R5**. For example, a C/C++ compilation process accesses header files and the `gcc` compiler binary. A proper combination of the filters can reveal these traces. Moreover, the compilation unusually touches a huge amount of header files, leaving peaks in the histogram, especially when performed in calm nighttime.

H.5.3 Clusters

Clusters (Figure H.0.1 – C) represent a generic mechanism enabling analysts to select files or directories with a specific "fingerprint". Clusters are defined by the combination of modification attributes (entries with *m-a-c-b* modification types) and regular expressions applied to the file names. Taking into account analytical requirements **R3** – **R5** and needs

of domain experts, we predefined several clusters covering the most common investigation tasks for UNIX file systems. Additional clusters can be easily appended.

- *All files* –The default cluster with no filtering.
- *User SSH files* – Configuration files and SSH keys stored in the users' home directories.
- *Standard executables* – Files stored in the standard system directories for binaries, e.g., `/bin`, `/sbin`.
- *Python/shell/PHP/perl scripts* – Several clusters based on standard file extensions, e.g. `.py`, `.sh`.
- *Cron definitions* – Files stored in the default locations of `cron` jobs, i.e., regularly executed services.
- *Starts with '.'* – Hidden files or directories.
- *Suspicious files* – Files or directories with names consisting of dots and white spaces.
- *Executables with sbit* – Executables that can run under a different user or group privileges than the original user or group.
- *Weak permissions* – Executable files writable for general users.
- *Compilation signs* – Access to C/C++ header files and the compiler executables.
- *Unusual commands* – Commands that are rarely used by common system administration, but often by attackers, e.g., `wget`, `curl`, and `shred`.
- *System configuration changes* – Important files related to the system configuration, e.g., `/etc/init.d` or `/etc/passwd`.

In the current implementation, only one cluster can be selected at one time. The number of all records fulfilling cluster criteria is shown as a "total entries" number. The "filtered entries" indicator shows the number of records satisfying other filtering criteria of the dashboard, and then they are listed in the *List view* and included in the *histogram*. A bar under each cluster box visually emphasizes the ratio between the filtered and total records, enabling the analysts to identify the impact of currently used filtering criteria on clusters.

H.6 System Architecture and Implementation

FIMETIS is designed as a client-server application. The client part is implemented as a web application built on the Angular framework. Interactive visualizations use the D3.js library. The server part provides services for file system data management (import, export) and interactive data processing via the client. The Flask REST API handles the client-server communication. Flask is a lightweight web server gateway interface written in Python, which mediates access to the backend API – the center of the application logic and communication with databases. This architecture enables a concurrent investigation of multiple sources. It is possible to open two file systems simultaneously in two different explorer windows, for instance, and explore them side by side.

Persistence (**R7**) is guaranteed by two database systems. The file system snapshots are

stored in the NoSQL Elasticsearch database. Configuration data, user accounts, interactions (e.g., bookmarks), and other operational data related to the analysis are stored in the rational Postgresql database.

H.7 Evaluation

To gather feedback on how well the tool fulfill the requirements **R1**–**R5**, and to identify possible refinements for the future design process iteration, we conducted a qualitative evaluation. The evaluation was held in June 2020.

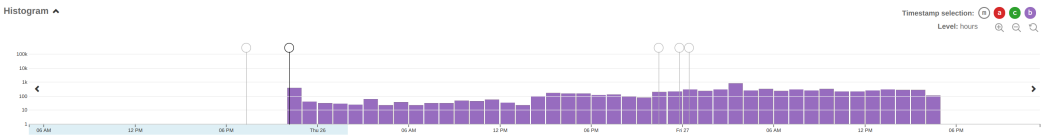

Figure H.7.1: Indication of a continuous creation of files generated by the network scanner.

H.7.1 Participants

We conducted the user study with five cybersecurity professionals who represent the target audience of the tool. All of them are members of the university cybersecurity research team or a security team in another organization. One participant works as an incident investigator in a private company. The average age of all participants was 30.2 years (*SD=3.5*); all of them were males. Two of them participated in initial interviews from which the requirements were derived. However, they did not participate on the design of the tool.

All the participants were cybersecurity professionals. However, they differ in the experience with practical investigation of incidents using file system analysis. Their skills are summarized in Table H.7.1.

ID	Age	Occupation	INC
P1	34	researcher in cybersecurity	<3
P2	32	researcher in cybersecurity	0
P3	32	incident investigator – network analyst	<3
P4	26	lead security analyst	>10
P5	27	incident investigator	>10

Table H.7.1: Demographic information of our participants. Occupation – position related to network administration and incident investigation, INC – number of incidents investigated by the analyst using disk analysis.

H.7.2 Data sets

During the evaluation, we used two datasets that were captured from computers affected by real incidents. The files were maintained using the *ext4* file system, which is commonly used on UNIX servers. We used different mechanisms to capture the primary data, yielding some records without the *b-time* timestamp (see H.4.1). The first dataset contained 308 311 records and was used for the tool demonstration and familiarization of participants with the dashboard. The second dataset consisted of 505 742 records and was used for the evaluation.

We carefully analyzed the second dataset using FIMETIS to reconstruct the incident to establish a baseline for the evaluation. Navigating through the predefined clusters, we gradually collected a list of crucial findings relevant to the incident. We identified six clusters that are most relevant to providing evidence of the incident.

- *User SSH files* – Displays access to SSH key files used by the attacker to control remote access to user's account.
- *Suspicious files* – A bunch of files is visible in `/var/tmp/...`. The directory name is suspicious (`...` is often seen during attacks) and it contained files named using IP addresses, suggesting it was used as a cache for network scans.
- *Executables with sbit* – In addition to standard Unix commands, the output reveals file `/var/lib/.s`, which is definitely not legit (tries to hide itself and elevates the executable rights using the root s-bit parameter).
- *Unusual commands* – Two HTTP command-line clients can be seen in the output that are used recently: `wget` and `curl`.
- *System configuration changes* – Changes to the machine user accounts can be identified in the output.
- *Compilation signs* – Several compilations of C-language codes are present in the dataset.

However, these pieces of evidence are often hidden in a huge amount of other entries. Therefore, using the list view and histogram is necessary to focus attention on relevant parts of the dataset. Having put all the collected information together, we compiled a precise summary of the incident and its timeline:

S1: 2016-05-25, 00:40: The attacker illegally logged in the account of user *martin* using SSH for remote access. Further analysis showed that the attacker abused unsecured NFS access to `/home` directory, allowing to upload of files and execution of privileged binaries. This is the only part of the analysis that could not be done just with the file system metadata, but the provided file system evidence gave a precise lead about what to check in the system logs and configuration.

S2: 2016-05-25, 02:40: The attacker installed a trojan code. A purportedly malicious `libselinux` library was downloaded using the `wget` command, and the system configuration (in file `/etc/ld.so.preload`) was changed to likely inject the library into

every newly created process. The SSH service was restarted to activate the trojan code (either a backdoor and/or credential-stealing). A suspicious s-bit file `/var/lib/.s` was installed simultaneously, probably to trigger the illicit activities.

S3: 2016-05-25, 19:20: There are suspicious activities in the account of user `roberto`. This account was probably also compromised a few hours later by the attacker as both the accounts show similar signs, e.g., an empty file named `1`. The reason is uncertain. However, there is no evidence that this account was used for suspicious activities.

S4: 2016-05-25, 21:22: The attacker re-compiled and re-installed the trojan code. The attacker was probably not satisfied with the version they deployed at the beginning of the day, so they returned, re-compiled the `libselinux` library, and then produced another binary on the spot.

S5: 2016-05-25, 22:08: The attacker created a hidden directory '`/var/tmp/...`', where they compiled some suspicious tools, e.g., `pcap` or `nmap`, and installed them into the system. Following that, they started a network scan and used the directory to store results obtained for individual network targets. Since then, the data was kept being captured and logged into this directory. The directory is used for a massive scan spanning almost two days, which is visible from the relevant histogram, see Figure H.7.1.

S6: 2016-05-26, 23:12: The system files with user account and passwords (`/etc/shadow` and `/etc/passwd`) were modified one day later. It is uncertain whether this activity is related to the incident or not.

H.7.3 Apparatus

The server part of the FIMETIS application was deployed on a common cloud machine, equipped with 8GB RAM, 80GB disk space and 4 CPUs. We conducted the evaluation online using Google Meet. The participants used Google Chrome on their computers or laptops with resolutions ranging from FullHD to UHD. Their interaction and comments were recorded for later analysis.

H.7.4 Procedure

The user study was divided into four parts. First, the participants were introduced to the general procedure, signed a consent form, and filled the demography questionnaire. Then, the experimenters presented the tool, explained all its features using the first dataset, and let the participant familiarize with the tool for 5–10 minutes.

Next, the participants were to find the following signs of the file system manipulation and usage:

T1: Files or directories with suspicious names.

T2: System files (configurations or executables) possibly modified by the attacker.

T3: Executables or libraries that were not installed from its package (i.e., either directly downloaded or manually compiled on the system).

T4: Privileged executables (with root s-bit) possibly used in the attack.

T5: Suspicious or unusual commands possibly executed by the attacker.

T6: Possibly compromised user accounts.

These tasks address requirements **R1**–**R5**. Together, they should provide an overview of what happened during the incident. While the tasks *T1, T2, T4, and T6* reflect different aspects of the detection of file system anomalies (**R3**), *T5* and *T3* are related to the execution of suspicious commands (**R4**) and traces of batch processing (**R5**) respectively. All the tasks require iterative exploration of the file system structure (**R1**) and temporal relationships (**R2**).

The participants had the tasks printed out so that they could easily make notes. The experimenter asked the participants to solve the tasks iteratively in any order. They were asked to think aloud. At the end of this evaluation phase, they had to summarize the incident upon their observations.

Although the real investigation of an incident lasts many hours or can even spread to several days, we restricted the participants to roughly one hour. The study's goal was not to get all the details about the attack, which is usually not possible without additional pieces of information such as system logs or network traffic, but to ascertain whether the analyst can get a quick insight into the incident using our tool.

When the incident investigation ended, the participant filled the usability questionnaire (Simple Ease Question, SEQ [203]), and System Usability Scale, SUS [202]. Finally, the experimenter interviewed participants on their final thoughts and feature requests.

H.7.5 Limitations

This user study has several limitations. The number of participants is relatively low. The reason lies in the time demands put on the evaluation process, which took roughly two hours per participant. To minimize the impact of this limitation, we involved security practitioners – possible users of the tool. On the other hand, we aimed to cover a wide range of expertise. Therefore, we engaged both highly skilled experts who have practical experience with collecting evidence from file systems and professionals who lack these specific skills as they focus on other cybersecurity domain, e.g., network analysis or cybersecurity research.

We are also aware that the evaluation was performed with only one test case, and then the results could be affected by the specific attack vector hidden in the dataset. We strove for

authenticity, and then we preferred a real incident from artificial data. On the other hand, we aimed to choose an incident which is typical in a sense. The selected dataset contains the digital evidence of common attack steps like the abuse of user accounts, privilege escalation, installation of backdoor, and using the compromised host for further illegal activities.

H.7.6 Results

Usability & learnability: User experience with the tool was evaluated by the System Usability Scale (SUS). SUS is a de facto standard method for assessing systems' usability regardless of their purpose. The average SUS score of FIMETIS was 88.5. According to the adjective ratings [18], the score corresponds to *excellent* ratings and proves compliance with **R6**.

SUS questions #4 (I think that I would need the support of a technical person to be able to use this product) and #10 (I needed to learn many things before I could get going with this product) can also be used to interpret learnability [202]. The average answers 1.2 and 1.8, respectively, on the Likert scale from 1: 'strongly agree' to 5: 'strongly disagree' suggest that FIMETIS is also easy to learn.

Preferences in using visual-analytic elements: FIMETIS is designed as a generic tool where hypotheses can be verified in various ways using the combination of diverse visual-analytical elements. To explore if some elements are more popular then other, we analyzed videos captured during the evaluation. We measured the usage of key interactions and data filtering concepts: filtering data by attributes, using predefined clusters, filtering data by span windows, searching and filtering by path, and using push-pins.

The results are summarized in Figure H.7.2. Push-pins represent the maximal number of bookmarks used by the analyst at the same time (20 push-pins in the participant P5). The other axes encode the relative time the analyst used the element. The time is expressed as the percentage of the investigation time. It is to be pointed out that the *name filtering* is used occasionally for temporal filtering and navigation during the interaction with the *List View*. Therefore, its usage can be underestimated in the radar charts.

The radar charts depicted show that different analysts preferred different combinations of elements. Usually, only 2–3 elements are used intensively, while others are ignored either completely or used significantly less. Another interesting observation, which is not captured in the radar charts, is that the analysts used only one span window. P1 did not use this element, and P3 used two span windows simultaneously, but only for a very short time.

Precision of the attack timeline: To evaluate the ability of the FIMETIS tool to provide a quick insight into the incident timeline, incident scenarios reported by participants were

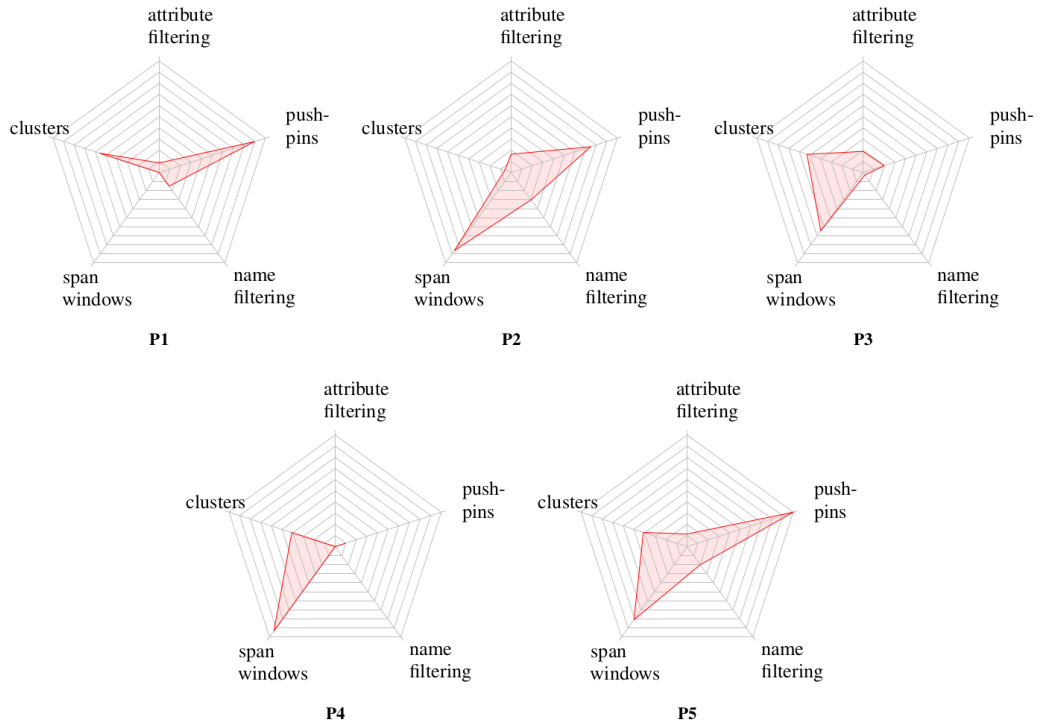

Figure H.7.2: Approximate utilization of visual-analytic elements of GUI by individual participants P1–P5. The *push-pins* axis encodes maximal number of bookmarks used simultaneously. Other axes represent the relative time (as the percentage of investigation time) when the element was used.

compared with the baseline scenario *S1–S6*. The precision was ranked by the authors of the paper. The results are summarized in table H.7.2.

	S1	S2	S3	S4	S5	S6
P1	●	◐	●	○	●	●
P2	●	●	●	○	●	●
P3	◐	◐	◐	○	●	●
P4	●	●	●	◐	●	●
P5	◐	◐	◐	○	●	●

Table H.7.2: Precision of the attack reconstruction: ○ overlooked/not identified, ◐ identified partially, ● identified correctly.

S1 (compromising the account 'martin') was identified by all participants. However, P3 and P5 identified the account together with 'roberto'. They did not decide who was the primary target of the attacker.

S2 (installation of a trojan code) was identified by all participants, but the level of observed details varied. All the participants discovered the `/var/lib/.s` as part of the attack vector, but P1, P3, and P5 did not provide more details about this attack phase. Moreover, the `selinux` library was completely overlooked by them. P2 did not mention the restart of the SSH server, but SSH was correctly identified as the service used for the escalation of privileges. P4 noticed and described all the details related to this attack phase, including the usage of `/etc/ld.so.preload`.

S3 (suspicious manipulation with the account 'roberto') was identified by all participants and considered part of the attack. Neither participant found the real abuse of this account. However, P3 and P5 did not decide whether the 'roberto' or 'martin' was the primary access point for the attacker.

S4 (re-compilation and new installation of the trojan code) was overlooked by all participants except P4. This analyst noticed the re-installation but overlooked the re-compilation of the trojan code at the compromised computer.

S5 (a hidden directory) was identified by all participants very quickly. The directory contained almost 12.000 records combining source code of multiple tools, traces of their compilation and usage, and data files gathered by the attacker. Nevertheless, the analysts were able to spot tools and data relevant to the attack vector and directly describe their purpose in the attack (P2, P3, P4, P5) or at least mention them as a tool worth further exploration (P1).

S6 (modification of the user account database) was identified by all participants. P1 noticed the changes but finally considered as not being linked to the incident. P2 did not provide more details. Other analysts considered the changes to be part of the attack when the attacker probably created a new user for later access.

Tasks difficulty: To evaluate the usability of the tool for solving individual tasks *T1–T6*, we analyzed the SEQ answers. We used this method because our tasks were too complex for metrics such as task duration time or completion rate, and the method performs as good as more complicated measures of task difficulty [203]. The participants responded to a single question associated with individual tasks ('Overall, how difficult or easy did you find this task?'), using a scale from 1 (very easy) to 5 (very difficult). The box plot is depicted in Figure H.7.3.

Overall, the participants considered tasks rather easy with the FIMETIS tool. This result correlates with the analysts' success to correctly reconstruct the incident in limited time at an appropriate level of detail. The only exception was finding out executables or libraries that were not installed from its package (*T3*). This task is considered rather difficult. However, this result also corresponds to the low success rate of revealing the re-compilation of a trojan code (step *S4* of the incident). The reason probably lies in the complexity of the task, which forces the analyst to iteratively combine multiple views and combine multiple

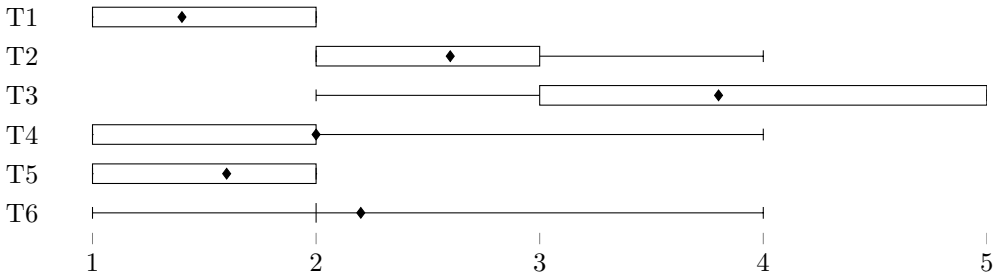

Figure H.7.3: Distribution of answers to SEQ tasks (min/max values, lower/upper quartile, and average). Lower score is better (1 = Very easy, 5 = Very difficult).

features of the tool.

H.8 Discussion and Future Work

The work we presented in this paper focuses on the design and user evaluation of a visual-analytics tool that aims to support efficient disk snapshot exploration as part of the cyber-security incident investigation workflow.

We collaborated with three skilled investigators on the clarification of forensic processes and the specification of requirements. The evaluation conducted with five cybersecurity experts revealed that the analytical tool built upon these requirements is intuitive and easy to use. All of the analysts were able to provide an incident report at surprising precision in very limited time. Moreover, it seems that the results obtained from less and more skilled analysts are subtle. We are aware that it could be affected by the attack vector of the incident selected for the evaluation, but this unexpected finding is promising for further development.

Another interesting observation was made regarding the usage of proposed visual-analytics concepts and their combinations. We noticed different workflows in using the tool by different analysts. This finding indicates that the tool is sufficiently generic. It supports various approaches to the verification of hypotheses and collecting the evidence. Moreover, the results captured in Figure H.7.2 suggest that there could exist a favorite combination of analytical elements. For example, the analysts P2 and P5 used predominantly span windows with name filtering and a lot of push-pins, while P3 and P4 preferred span windows and clusters combined with only a few push-pins. Exploring such behavioral patterns would bring insight into analytical strategies. However, it requires a much deeper evaluation and analysis in future work.

Our work is still in progress. During the user study, we collected user feedback and requests for additional useful features.

File system attributes management: Multiple analysts forgot to cancel the per-attribute filtering during the investigation. This mistake led to false hypotheses and delay in the investigation. Emphasizing this filter or indicating that the *List View* contains only entries with selected modifications are required.

Dealing with file system records: The *List View* is the primary source of information for investigators, and efficient manipulation with records has shown to be the key factor for the investigation process. In spite of searching, filtering, and smart navigation techniques implemented in the *List View*, the analysts requested even more features for rapid navigation in the list. Especially, scrolling the list to a record by `CTRL+F` hotkey was missing. Currently, only highlighting and filtering out the data by the typed text is implemented in the tool. Also, the support of regular expressions and hiding records matching the typed text temporarily were required. Complementary hierarchical views to the strictly temporal ordering of records, e.g., using treemaps to convey space requirements of file system parts, reveal anomalies, and navigate to them quickly, will be considered in the future work.

The current implementation of FIMETIS serves as an analytical and decision-making tool for file system metadata analysis (Figure H.1.1). Although the evaluation proved the usefulness of the tool, users ask for the support of other parts of the investigation process as well. Reaching this goal requires making significant extensions to current functionality and then to the design. In what follows, we outline key requirements and their possible impact on visualizations and GUIs.

Incident report creation: Incident reports are key outputs of the investigation process. As a lot of clues and pieces of the incident evidence appear during the interaction, it would be useful to use them for the report creation. Apart from online notes that have already been integrated into the new version of FIMETIS, investigators' feedback revealed possible changes in using bookmarks for this purpose. Currently, bookmarks are very simple. They are represented as push-pins referring to interesting records (points in time) and used for fast navigation (jumping to these records). Multiple analysts were asking for the possibility to distinguish between push-pins by color, tagging them, and making their own notes. Once the concept of bookmarks is moved from push-pins to advanced annotations, it would be possible to use them for the direct generation of incident reports or their parts.

Analysis of system logs: File system metadata represents only one source of information for investigators. Other data sources, like system logs or network traffic data, are often available to provide a broader context. Especially so-called super-timelines, i.e., file system metadata merged with system logs, are often used for forensic investigation. Extending FIMETIS with system logs should be possible. Both types of data sources are time series. The proposed approaches to file system exploration seem to be reusable also for system logs. However, further research and evaluation are needed. It is especially necessary to balance between unified exploration, when an analyst uses both data types together, and distinguishing both contexts as they represent different knowledge with possibly different uncertainty.

Other information sources: Ability to analyze other data sources like network traffic or memory snapshots are required by forensic investigators as well. However, they encode very different data with very different abstractions that require the application of specific visual-analysis techniques and concepts. Therefore, narrowly focused tools are designed that provide comprehensive visual-analytics interfaces [42]. Joining these information sources into a single "silver bullet" analytical tool can be counter-productive and going against the **R6** requirement.

We aim to address the aforementioned features and enhancements in future work. As the FIMETIS application is already used in practice for the investigation of real-world incidents (three incidents were successfully investigated by the security teams of Masaryk University and CESNET so far), we aim to utilize this experience to extend the functionality of the application further. Especially, we plan to introduce advanced user-defined clusters and the support of multiple timelines, e.g., records of system logs. These extensions will require changes in the current design and the development of new visual-analytic methods to cope with even bigger and more variable data.

Acknowledgment

This work was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

Article I

Visual Feedback for Players of Multi-Level Capture the Flag Games: Field Usability Study

Radek Ošlejšek[1], Vít Rusňák[2], Karolína Burská[1], Valdemar Švábenský[1], Jan Vykopal[2]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2] Masaryk University, Institute of Computer Science, Brno, Czech Republic

VizSec – IEEE Symposium on Visualization for Cyber Security. 2019, 11 pp.

Abstract

Capture the Flag games represent a popular method of cybersecurity training. Providing meaningful insight into the training progress is essential for increasing learning impact and supporting participants' motivation, especially in advanced hands-on courses. In this paper, we investigate how to provide valuable post-game feedback to players of serious cybersecurity games through interactive visualizations. In collaboration with domain experts, we formulated user requirements that cover three cognitive perspectives: gameplay overview, person-centric view, and comparative feedback. Based on these requirements, we designed two interactive visualizations that provide complementary views on game results. They combine a known clustering and time-based visual approaches to show game results in a way that is easy to decode for players. The purposefulness of our visual feedback was evaluated in a usability field study with attendees of the Summer School in Cyber Security. The evaluation confirmed the adequacy of the two visualizations for instant post-game feedback. Despite our initial expectations, there was no strong preference for neither of the visualizations in solving different tasks.

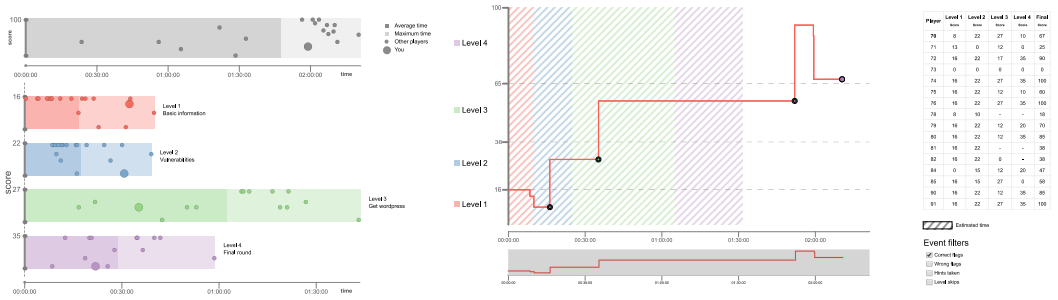

Figure I.0.1: Clustering (left) and Timeline (right) interactive visualizations provide feedback to players of serious multi-level cybersecurity games. They offer holistic as well as person-centric perspectives on the score development of one or more players.

I.1 Introduction

As cyber attacks have been on the rise in recent years, security professionals and students have to be trained in adversary thinking, which enables them to understand cyber attacks and set up effective defenses. A popular way of cybersecurity training is through gamification and serious games [7, 160]. A general shortage of methodologies and tools for timely feedback in the field of serious games is emphasized in [15, 22]. This deficiency is even more apparent for cybersecurity serious games, which pose specific demands on environment capabilities [222].

The subject of our research is to provide meaningful insight into the training progress. In this paper, we investigate how to provide valuable visual feedback to players of serious cybersecurity games right after the exercise so that they can immediately learn from their experience and compare their results with other players. Following the terminology used in the cybersecurity training domain, players of serious games are referred to as *trainees* in this paper.

Serious games are of many types. To reach the goal, we restrict ourselves to Capture the Flag (CTF) games [251, 33, 239] that are played in virtual environments, in which gameplay events can be monitored and used for providing automated visual feedback to trainees.

This paper deals with a multi-level variant of CTF games. Each game, regardless of its specific objectives or content, consists of well-described tasks divided into consecutive levels. The access to the next level is conditioned by fulfilling tasks from the previous one. Moreover, players can take hints or skip the entire level. Points are awarded or deducted for these actions so that the final scores of individual players are mutually comparable and can be used for their evaluation.

The design and development of our feedback visualizations went through four stages: (1) understanding of serious multi-level games, their objectives in cybersecurity domain, and available data, (2) defining requirements on the visual feedback in accord to the educational

goals of the games, (3) prototyping, iterative design and development, and (4) usability study performed at our university and involving students of our cybersecurity training lessons.

Contributions. The main contributions of the paper are: (a) we classified the visual feedback requirements into three categories that cover trainees' expectations of the training (personalized feedback, comparative feedback, and overall results); (b) we applied existing visualization techniques in the domain of hands-on cybersecurity training in order to provide better insight into the trainees' results right after the training session; (c) we performed a formal evaluation that confirmed the meaningfulness of the defined requirements and usefulness of the post-game analysis visualizations for trainees.

The remainder of the paper is organized as follows: Section I.2 introduces the related work in the area of visual analysis of serious games, particularly in the cybersecurity domain. Section I.3 describes available data and provides the example of a cybersecurity game. In Section I.4, we formulate requirements posed on visualizations and corresponding tasks covering three different cognitive perspectives on game results. We discuss our approach to fast visual feedback in Section I.5. Section I.6 describes the usability study and brings necessary details about the usability testing concerning defined hypotheses. Results of the usability study are discussed in Section I.7. We draw our conclusions and look to the future in Section I.8.

I.2 Related Work

Many works published in the field of user behavior visual analysis focus on social media. Cao et al. [41], for instance, proposed a visual analysis tool for anomalous user behavior in online communication systems and social media platforms. The proposed system incorporates presentation of user's communication activities and interactions. Another analysis-driven approach to user behavior was introduced by Kumar et al. [134]. They investigated users' migration behaviors among various social media platforms and represented the findings via radar charts. Many other works also address the topic of social media-related user behavior in varied scopes, such as [257, 225, 40]. A comprehensive survey on visual approaches to the analysis of anomalous users can be found in [213]. Our solution also supports behavioral analysis but in a very simplified way enabling players of CTF games to learn from their behavior by being aware of their steps and steps of other players.

The positive effects of visual analytics integration into the learning process have already been identified. The outcomes serve for understanding trainees' actions or optimization of the learning environment [232, 142]. As the educational visualization dashboards have gained considerable attention in the field of learning analytics, reviews concerning this matter emerged as well [31, 209]. The visualizations can monitor trainee's progress and help to compare the performance with other peers [93]. They can also increase motivation and encourage trainees to compete or to collaborate [94]. In this paper, we focus on automatically

generated post-training feedback. In [55], the authors investigated how students interpret feedback delivered via learning analytics dashboards in distinct courses. Their findings reveal that the majority (83%) of students were able to identify gaps in their performance.

Various general tools for qualitative feedback or assessment in education exist, and many studies include a platform, where the dashboards are employed in multiple field of education. They serve for capturing the behavior of students or provide long term statistics and observations for both students and teachers [120, 61, 138, 151]. An online tool *asTTle*, for instance, can assess students' achievements and progress. It allows creating pen-and-paper tests, and then continuously analyzing specified characteristics, which can be stipulated by teachers. The students' outcomes are then visualized in interactive reports that provide rich feedback related to student performance [105]. *Questionmark Perception* [188], another example of a similar system, is used for education in the form of surveys, tests, or exams and enables the creation of reports from these events. Govaerts et al. [94, 93] proposed a general-purpose web-based environment for the visualization of the students' progress and results based on the tracking and evaluation of Twitter hashtags. Their tools help the students to assess themselves and to get automated feedback on their achievements throughout an online course.

These works, however, often tend to take a 'one-size-fits-all' approach to the collection, processing, and reporting of data, overlooking disciplinary knowledge practices. Furthermore, since they focus on long-term courses, they are not suitable for our needs as they require different perspectives on the analysis of the learning process. While the useful applications of visual analytics in education are well known, we lack its use in the cybersecurity training. Therefore, as there are learning dashboards designed for teaching specific disciplines like programming [99, 87], mathematics [118], or specific cyber defense exercises [242], we aim at providing learning feedback in a specific domain – CTF games.

A comprehensive list of numerous cybersecurity CTF games can be found at the CTFtime portal [57]. However, most of the listed platforms provide only limited information about collected data and their presentation to the users.

CTFd [53] is a platform for creating and hosting CTF challenges. It visualizes overall score graphs and breakdowns for individual players. The latter includes percentages of all challenges solved, distributions of solved challenges into categories, and the evolution of player score over time. However, user evaluation of the platform effectiveness is missing.

EDURange [249] allows gathering the command-line history of a trainee, including command timestamps, their arguments, and exit statuses. The platform can automatically generate an oriented graph that visualizes the command history. The vertices of the graph represent executed commands; the edges represent the sequence of commands (that is, an edge from a command x to y means that y was executed after x). Supervisors can use the graphs in real time to check how the trainees progress and whether they need extra guidance. A post-game use case would be to compare the graphs to each other or a previously

prepared pattern of a sample solution.

During a *Crossed Swords* exercise [127], network traffic, logs, and system activity metrics were collected and analyzed. The goal was to provide real-time feedback and situational awareness for the trainees. In a post-game survey, 11 out of 14 participants found the feedback useful for their learning, and the remaining three were neutral. The authors raise the question of finding a balanced amount of information to provide to participants since 4 of them reported being distracted by the feedback.

PicoCTF is an online competition that targets high school students. The competition comprises a series of challenges in the form of an interactive storytelling game [262]. An evaluation of the game design is based on survey responses and user interaction logs [47]. The collected data is not used for assessing the players.

An international *iCTF* competition provides feedback in the form of a scoreboard, which is also used during the game to inform the competitors of the score development and status of their services [67].

From the information available, most of the CTF platforms offer only a simple scoreboard for comparing players' final score. Techniques used in current security training programs do not facilitate any further summative assessment or feedback for the players regarding their actions in the game [201, 169].

I.3 Dataset Characteristics and Game Example

This section describes the data collected during multi-level CTF events. The introduced game example explains available data in detail and demonstrates the principles of cybersecurity CTF games. Moreover, the presented game was selected for our usability study.

I.3.1 Selected Cybersecurity Game

We chose a cybersecurity game named *The Biggest Stock Scam Of All Time*. The game was created by the students of Cyber Attack Simulation course [239] and was further improved by cybersecurity experts. In a background story that complements the game, the trainee takes on a role of a former employee of a global stock trading company. However, he was fired because of refusing to falsify the reports of the company's earnings. When someone else did the job, he wanted to prove the company's corrupt intentions, but to gain evidence, he needs to access the company's records.

Figure I.3.1 shows the game's network topology. At the beginning of the game, the trainee receives control of a single attacker virtual machine in a realistic environment emulated by the KYPO cyber range [243] The machine runs Kali Linux, a standard distribution for penetration testing. The learning objective of the game is to practically exercise cyber

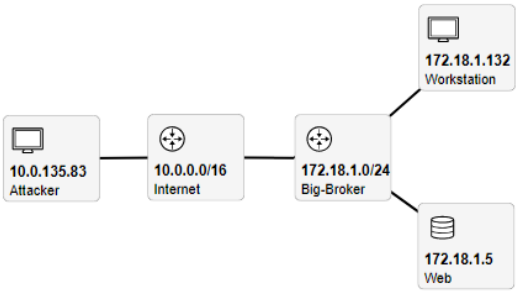

Figure I.3.1: The trainee's view of the game network topology.

attacks. In four consecutive levels, the trainee must gain access to the company's web server, which is available only from an exposed workstation in the company's internal network.

1. In the first level, the trainee learns about the existence of a company's workstation that is accessible from the internet. An `nmap` scan reveals that TCP port 22 is open on the workstation, running the SSH service. Having a list of common usernames and passwords, the trainee performs a dictionary attack and accesses the workstation.

2. From the workstation, the trainee can now access the server, which hosts a WordPress website. Since old versions of plugins in WordPress websites are known to contain vulnerabilities, the trainee scans the server using `wpscan` and reveals a file upload vulnerability.

3. The trainee proceeds to exploit the vulnerability using *Metasploit* penetration testing framework to gain shell access on the server. However, the trainee does not yet possess the necessary privileges to read the file with the stock dealings.

4. Although the server is well-secured, it allows to run `tcpdump` as a root user, which allows executing scripts. Thus, the trainee escalates the privileges on the file and reads the company's dark secrets.

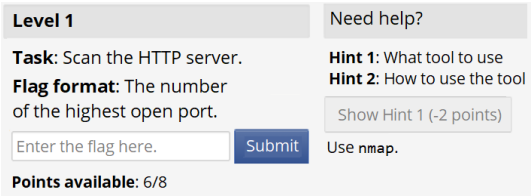

Figure I.3.2: The trainee's view of the game interface.

Figure I.3.2 shows the game's user interface. Completing each level yields a flag worth a certain number of points that add up to the trainee's score. Each level also contains hints, which the trainee can view in exchange for a scoring penalty. If the trainee is still stuck,

(s)he can display the full solution to the level but will not receive any points for it. This scaffolding system allows each trainee to progress individually.

I.3.2 Data Logging Explanation

We record the interaction with the game interface in the form of *game events*. There are eight game events: starting the game, ending the game, starting a level, ending a level by skipping it, ending a level by submitting a correct flag, submitting an incorrect flag, taking a hint, and displaying a solution to the level.

Each game event is logged as one line in a CSV file[2]. with the following five-part structure:

- `player_id` – a unique numerical ID randomly assigned to each trainee before the game,
- `timestamp` – absolute time in the format YYYY-MM-DD HH:MM:SS,
- `logical_time` – relative time from the start of the level in the format HH:MM:SS,
- `level` – the order of the level,
- `event` – one of the eight game events described above.

An example of a record in the log is:

```
9003581,2018-08-24 16:57:54,00:03:42,4,Hint 3 taken
```

I.4 User Requirements

Content of CTF games differs. It is therefore impossible to predict and conceptualize content-related questions that trainees would be interested in as part of the post-game feedback. Instead, we focused on capturing high-level user requirements that follow the unified structure of multi-level CTF games and corresponding data. To elicit the requirements, we organized discussions with four domain experts who regularly organize CTF games and who understand educational aspects of training. These experts are skilled in providing informal feedback to players right after the training session and then they have insight into interests of trainees during the post-training debate. Two of the experts are co-authors of this paper. Based on discussion with the experts, we defined three high-level requirements for the visual feedback that helped us to conceptualize views on game data and to design specific visualizations:

***R1:* Provide personalized feedback.** Players should be able to find out their results and identify their well-done and problematic parts of the game. This requirement includes

[2]Supplementary materials also available at `https://www.radek-oslejsek.cz/it/supp-materials/`

person-centric goals and questions regardless of other players, e.g., "In which level did I lose the most points and why?".

***R2:* Provide comparative feedback.** Players should be able to identify parts of the game where they were better or worse than other players. This requirement introduces a competitiveness factor into the feedback, which enables players to compare themselves with others and assess their abilities within a group, typically in a competition.

***R3:* Provide a brief overview of the overall game results and features.** Players should be able to get a necessary insight into the game difficulty and other aspects that enable them to put their personal and comparative findings into the context of this particular game. It is useful mainly in situations when a user plays multiple games. In such a case, the user might want to know why he or she was more successful in one game than in the other, for instance. After group-based training sessions, users are often wondering who was "the best player" in the group so that they can further explore his or her tactics and behavior. However, "the best" is not easy to define, as seen in tasks **T11** and **T12** that deal with various views on "to be the best".

Requirements *R1–R3* delimit the design of interactive visualizations and provide an initial classification for possible interactions. Additional constraints ensue from the available game data. Altogether, they have been considered during the design process (our assumptions on datasets are described in detail in Section I.3). However, there is still a variety of options for a suitable solution.

To specify user requirements more precisely, we refined *R1–R3* into particular interactive tasks that are summarized in Table I.4.1. We aimed to cover various aspects of the high-level requirements. To reach this goal, we built on the analysis of the data available from previous training sessions and the discussion with domain experts who iteratively commented on proposed tasks and voted for them. The resulting list of tasks was, therefore, reached as the consensus of the four aforementioned domain experts. Real meaningfulness of the tasks for players was then verified along with the evaluation of the designed visualizations, as discussed in Section I.6. We are aware that there can be many other possible tasks associated with the requirements. However, as the goal of this study was to design and validate initial tool serving as an automatically generated visual feedback, we consider the tasks representative.

In addition to supporting user requirements *R1–R3*, one more critical feature had to be considered: The visual feedback has to be intuitive and easy to use since the reflection phase following the training session is often very short (several minutes only), and providing complex visual analytics systems would be counter-productive. The balance between providing a complex set of information, relevant to the trainee and the feedback simplification pose a challenging task.

Table I.4.1: Interactive tasks covering requirements *R1–R3*.

R1	T1:	Find out when you finished the game.
	T2:	Find out in which level(s) you reached the lowest score.
	T3:	Find out your final score.
	T4:	Find out how much time you spent in the 2^{nd} level.
	T5:	Find out when you advanced from $2^{nd}$to 3^{rd} level.
	T6:	Find out in which level you lost most points in the game.
R2	T7:	Characterize your score compared to other players.
	T8:	Characterize your time spent by playing compared to other players.
	T9:	Find out the player who reached the closest score to your score.
R3	T10:	Find out how much time was assigned for playing the game.
	T11:	Is there somebody who reached a high score in significantly short time? Insert his ID.
	T12:	Find out who reached the best score.

I.5 Visual Design

We designed two complementary visualizations to provide visual feedback to the trainees of multilevel games. They combine a known clustering and time-based visual approaches to show the score and its development over time in the way that is easy to decode. At present, the visualizations are independent, and each serves for a bit different purpose, presents the data from a different perspective, and on a different level of detail. They were designed to cover together requirements *R1–R3*.

I.5.1 Clustering Visualization

The first view on the recorded data is presented in the form of a bar chart visualization combined with a scatter plot, as shown in Figure I.0.1 (left). It exploits a clustering principle to demonstrate achieved score results and bar chart principles to show time requirements. When designing this visualization, we emphasized the simplicity so that trainees can quickly focus on retrieved score and get a fast overview of their results (*R1*), results of other trainees (*R2*), and the overall distribution of results in the game (*R3*).

The visualization is split into two parts. The upper part with a gray bar includes the overall results of the game, while underneath there are results from individual levels.

The length of each bar expresses the maximum time for the given level (i.e., the time of the slowest trainee). Bars and timeline on the x-axis are scaled automatically according to the recorded timestamps so that the chart fills the canvas regardless of the game duration. The brighter shade denotes an average time of the level or the game, respectively. Trainees whose results appear in the darker part were faster than the average of all trainees and vice versa. The height of the bars is fixed, although the scoring span can differ in each level. Instead, the scoring span is indicated by numbers next to the y-axis. The fixed height

enables users to attract their attention to relevant results in each level or the whole game.

Results of individual trainees are displayed as small dots. Their vertical and horizontal position in the bars corresponds to their score and time. To support the person-centered view, one of the dots that represents the current trainee is always bulkier than the others. When the mouse pointer hovers over the dot, the corresponding dots of the trainee are highlighted too. This helps to keep track on the individual scores and times of the inspected trainee across multiple levels. Simultaneously, the exact time and achieved score are displayed, as portrayed in Figure I.5.1. The visualization implements a snapping functionality for attracting the mouse pointer towards the dots to make their selection more comfortable.

Clusters of points can be used to identify the correlation between time and score visually. Horizontal clusters of points, for instance, reveal users who obtained similar score while vertical clusters show users who spent a similar time in the game levels.

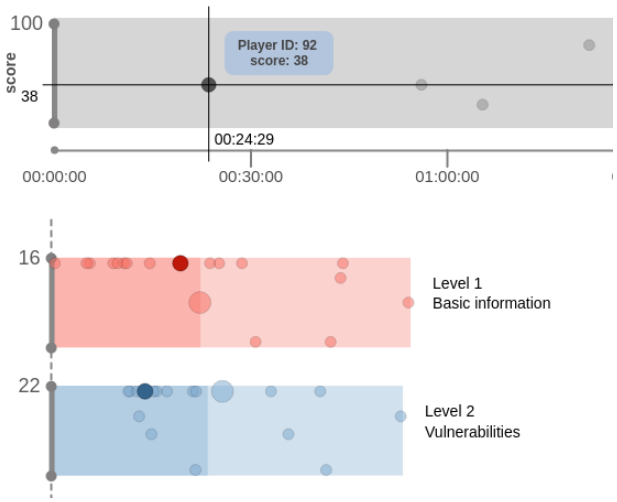

Figure I.5.1: Clustering detail with a selected point on the game overview bar and highlighted corresponding results in level bars.

I.5.2 Timeline Visualization

The second visualization, shown in Figure I.0.1 (right), demonstrates the progress of trainees throughout the game. It is more time-oriented than the previous clustering visualization and also contains more details from the gameplay, including details about hints and penalties that can be shown on demand.

On the x-axis, there is a timeline, while the y-axis captures score values. The horizontal dashed lines indicate a maximal number of points reachable in the game. In Figure I.0.1, there is at most 16 points after the first level, 38 points after the second level, 64 points after the third level, and 100 points overall. In contrast to the clustering visualization

where the bars are based on recorded timestamps from the game, the striped background of the graph outlines an estimated time for each level of the game.

Polylines in the visualization, further referred to as *scorelines*, represent fundamental graphical elements showing the development of achieved score of individual trainees. Upon entering a new level, the score increases with the maximum point value for the level and then the *scoreline* significantly "jumps up" at this moment. Upon gaining a penalty for providing a wrong flag, taking hints, or skipping the level, the *scoreline* decreases proportionally. Marks of specific events can enrich the scoreline. A pop-up tooltip with event details raises when the mouse cursor hovers the mark, as shown in Figure I.5.2. Types of events to be shown are controlled by the selection filter next to the main view.

Events in the *scoreline* can be dense. Therefore, we integrated a zooming function into the chart. After zooming, the chart under the main graph provides an overview of zoomed time span and enables the user to adjust a cutout and shift the time span easily.

To support person-centric tasks, the *scoreline* of the current trainee is emphasized. Score lines of individual trainees can be turned on and off by clicking in the adherent table. They have assigned different colors to distinguish score lines of different trainees. The color mapping is shown in the table (color stripes in rows 70 and 71 in Figure I.5.2, for instance) so that the trainee keeps track of the relationship between table rows and score lines.

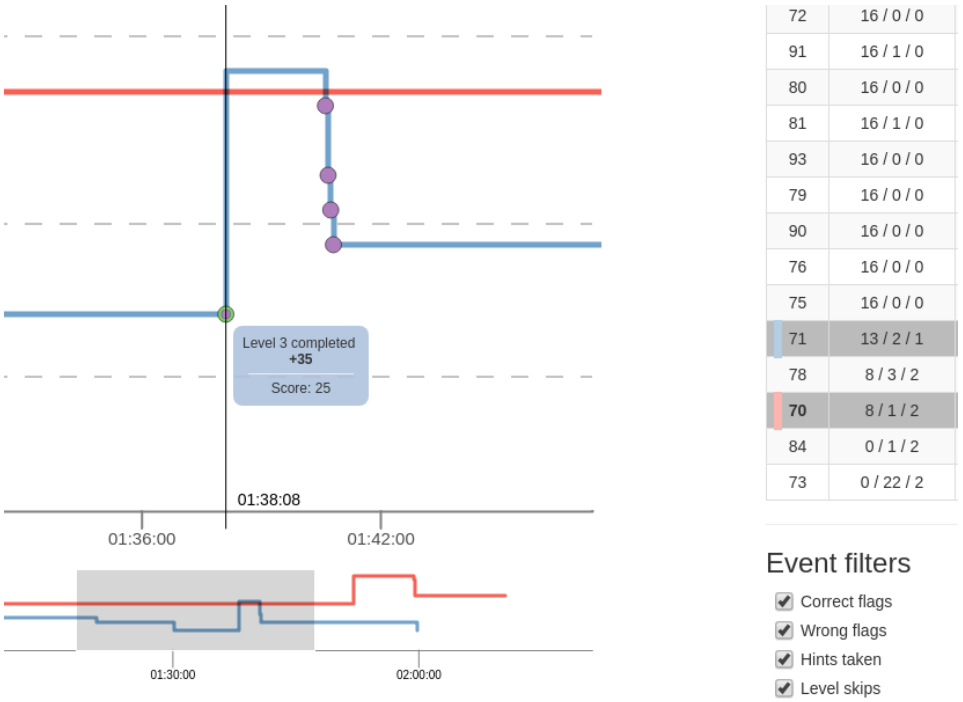

Figure I.5.2: Timeline detail: Zoomed score lines of two trainees with pop-up tooltip of selected event.

The interactive table also helps in a detailed exploration of results. It provides information about exact scores in individual levels as well as the total score. Users can also quickly compare results by sorting rows according to the selected column (level or final score).

I.5.3 Implementation

Both visualizations are implemented as Angular modules using the D3.js library for drawing. A demo version with test data is available online at `http://kypo-summer.surge.sh/`. The visualizations are adapted for independent testing and evaluation. Although they contain a dummy dataset (which is different from the data used for this usability study), it is also possible to upload new game definitions (number of game levels, their names, etc.) and corresponding logs capturing the gameplay data of trainees. The game definition and event logs used in this usability study are available in supplementary materials [2].

We are currently integrating the visualizations into our KYPO Cyber Range [243] – a cyber exercise and research environment that is used for the organization of hands-on cybersecurity training [239]. When the integration is done, the data will be available at runtime, and the visualizations will become a part of the learning process.

I.6 Usability Study

In this section, we describe the details of the usability study we held to evaluate the usability of the visual feedback. We decided to conduct a within-subject user evaluation with the attendees of Summer School in Cyber Security 2018 to mimic a real-world use case of the visualizations. The user study was included as a part of the program. Participants were finalists of high-school Cyber Security Competition 2018.

I.6.1 Hypotheses

We have formulated three hypotheses for the evaluation. They address the meaningfulness of our user requirements and related tasks, the usefulness of the visualizations in solving the tasks, and the identification of strengths and weaknesses of our approach. They are defined as follows:

H1: Requirements *R1*–*R3* are meaningful and useful for trainees. User requirements and their corresponding tasks were distilled from the discussion with domain experts – game designers and organizers of CTF games. The goal of this hypothesis is to verify that the requirements are also meaningful and useful for trainees and that they sufficiently cover their interests. To verify this hypothesis, the players rated the meaningfulness of individual tasks.

H2: The visual feedback is useful in providing insight into the tasks of *R1–R3*. This hypothesis should verify our assumption that the visual feedback provides straightforward and easy to decode way for seeking relevant information. Verification of this hypothesis was based on the qualitative and quantitative evaluation, during which the participants solved tasks and rated their difficulty.

H3: Some visualizations or their parts are more useful for specific tasks of *R1–R3* than others. The Clustering and Timeline visualizations were designed as complementary, providing different views on the data with different level of detail. However, most tasks **T1**–**T12** can be solved by both of them, in an easier or more difficult way. The hypothesis H2 should uncover the usefulness of the visual feedback as a whole, regardless of which visualization was used to solve the task. On the contrary, this hypothesis aims to verify whether some views or parts of the visual feedback fit better to some tasks or user requirements than others. Our goal is not only to confirm or reject the hypothesis but to identify such tasks and visualizations. To reveal this type of information, we asked for the usefulness of individual views for solving particular tasks.

I.6.2 Participants

Out of 16 attendees of the summer school, 12 senior high-school students (1 female, 11 male) participated in the study. All of them were between 16 and 19 years old, with normal or corrected-to-normal vision. None of them was color blind. All of them were daily users of smartphones and computers (laptops or desktops). Nine of them considered themselves as experienced users familiar with cybersecurity topics, one as a complete newcomer, one as a novice user, and one as a security professional. Although none of them was a native English speaker, they were proficient enough to understand the English questionnaire.

I.6.3 Environment

The evaluation was conducted on all-in-one computers with FullHD displays (1920 × 1080 pixels) running Windows 10 and the latest stable Google Chrome as an internet browser. The browser was maximized the whole time. The participants used the same computers as in the previous activities of Summer School.

We used the LimeSurvey online questionnaire tool for presenting the informed consent form, instructions, task assignments and complementary questions (task meaningfulness, difficulty, and visualization preference for each of the two). The tasks were organized one per screen and accompanied with supplementary questions.

I.6.4 Procedure

The participants engaged with the CTF game described in Section I.3. They utilized a web interface of the KYPO Cyber Range [243] to play the game. The feedback visualizations were not available to players during the game. The rough time assigned for the CTF game was 90 minutes. When the time ran out, the participants who did not finish had to terminate the game using the "Skip level" function.

Then, there was a 20-minute refreshment break during which the operators prepared the LimeSurvey questionnaire, and set up the feedback visualizations. The latter step included loading real logs from the CTF game. The questionnaire and the two visualizations were opened in separate tabs within the Google Chrome browser. Since CLUSTERING and TIMELINE visualizations were designed as complementary, the participants could use both to accomplish the tasks. Therefore, the participants had to switch between tabs when they solved the tasks. The names of the visualizations were also displayed in tab labels to avoid unintentional terminology mismatch between them.

After the break, the operators explained the purpose of the experiment and the user study procedure comprising of three parts. We explicitly asked the participants not to collaborate among each other.

First, the participants were introduced with the two visualizations, and they had up to 10 minutes to familiarize with both of them. Next, the 12 tasks described in Table I.4.1 were assigned to them. To mitigate the ordering bias, we randomized the order of the questions for each participant. Last but not least, the participants answered several demographic questions. In total, we reserved 50 minutes for the user study. One of the operators was present the whole time to provide support with technical issues.

I.6.5 Results

This section presents the result of a quantitative and qualitative evaluation of the data acquired from the usability study. The questionnaire and collected answers are included in supplementary materials[2].

The independent variables included in the study were tasks (12) and visualizations (2). For these, we measured two dependent variables: task correctness and ordinal data from 6-point Likert scales focused on usability of each visualization for a particular task (1 = Absolutely useless, 6 = Absolutely useful), difficulty of the task and its meaningfulness (both: 1 = Strongly disagree, 6 = Strongly agree), as discussed in what follows. We obtained 192 trials (12 participants × (12 tasks + task meaningfulness + task difficulty + visualization preference)) from the usability study.

H1 – Meaningfulness of Requirements

To verify that the design requirements *R1*–*R3* were chosen reasonably and the tasks **T1**–**T12** reflect users interests, we analyzed answers to the question *The task was meaningful* that has been asked after each task. Figure I.6.1 presents the median and mode values. The overall score provided by participants is positive, however, not significantly. $Median = 4$ (= somewhat agree) for all three requirements. $Mode = 5$ for *R1* and *R3*, $mode = 3$ for *R2*.

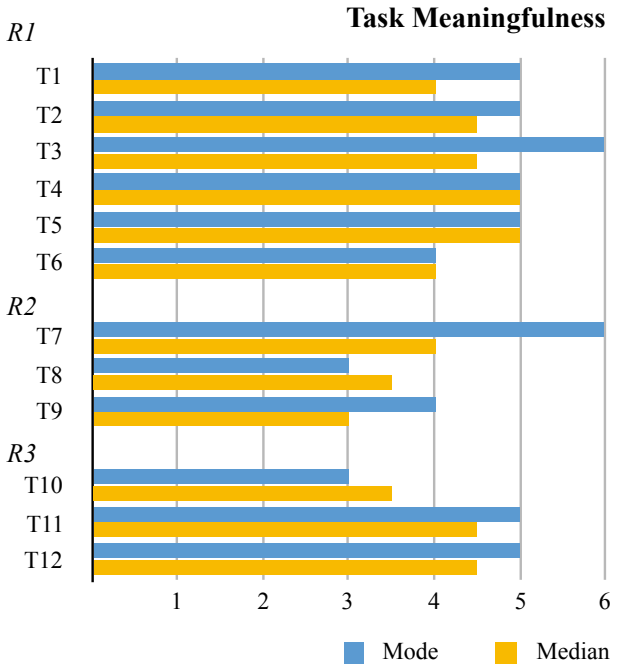

Figure I.6.1: Evaluation of the tasks meaningfulness. *Question: The task was meaningful.* Higher score is better (1 = strongly disagree, 6 = strongly agree).

We can notice higher scores in case of *R1* and *R3* compared to *R2*. It means that participants considered tasks in these two categories more meaningful when reflecting their gameplay. Friedman test reveals no statistical significance ($\alpha = .05$) among the tasks within the same requirement group: *R1* ($\chi^2 = 4.97, df = 2, p = 0.42$), *R2* ($\chi^2 = 4.96, df = 2, p = 0.084$), and *R3* ($\chi^2 = 1.51, df = 2, p = 0.459$). This observation confirms hypothesis H1 and our assumption that the tasks reflect users interests.

H2 – Usefulness of Visualizations

Both CLUSTERING and TIMELINE visualizations were designed as complementary. Therefore, our goal was not to compare usefulness head-to-head but evaluate their usefulness for completing the tasks.

First, we analyzed every task individually to determine (in)correct responses and their ratio. Since the data was produced by participants while playing the game, thus are not synthetic, some of the tasks do not have a simple one-value correct solution. Moreover, in time-related questions, there can be inaccuracy in answers caused by many reasons, e.g., approximate mouse location on the visualization, ignoring seconds by the user, etc. Due to these facts, we checked individual responses and categorized them into three groups: wrong, correct, and partially correct. This assessment was reached as the consensus of the authors of the paper.

Figure I.6.2 presents resulting (classified) responses. The overall combined success rate (including both correct and partially correct responses) is 73 % with eight correct responses per task on average. Therefore, we can conclude that the trainees were successful in performing tasks in general.

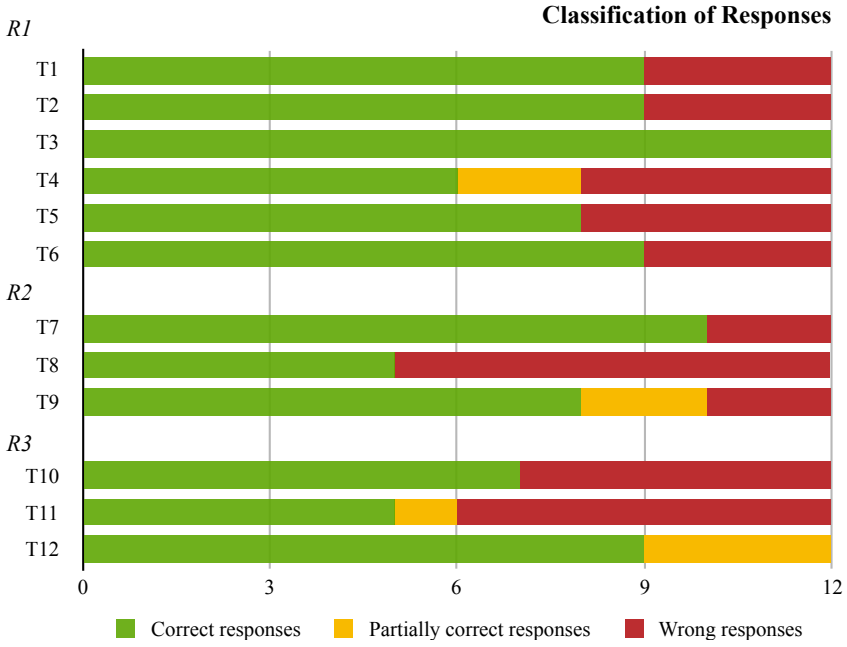

Figure I.6.2: Responses classification for each task.

We identified four occurrences of partially correct responses. The imprecise answers when the reported time was rounded to the nearest minute (e.g., 0:15:00 instead of 0:14:59) in **T4**. In **T9** and **T12**, there are incomplete answers when participants reported only a subset of multiple correct answers (e.g., two participants reached the same best score but only one of them was reported). In **T11**, it was due to unclear data when there was no strong evidence with multiple correct options.

Figure I.6.3 depicts the results of the qualitative evaluation of the Hypothesis H2 presented by the question *The task was easy to complete (using the visualization)*. Regardless of

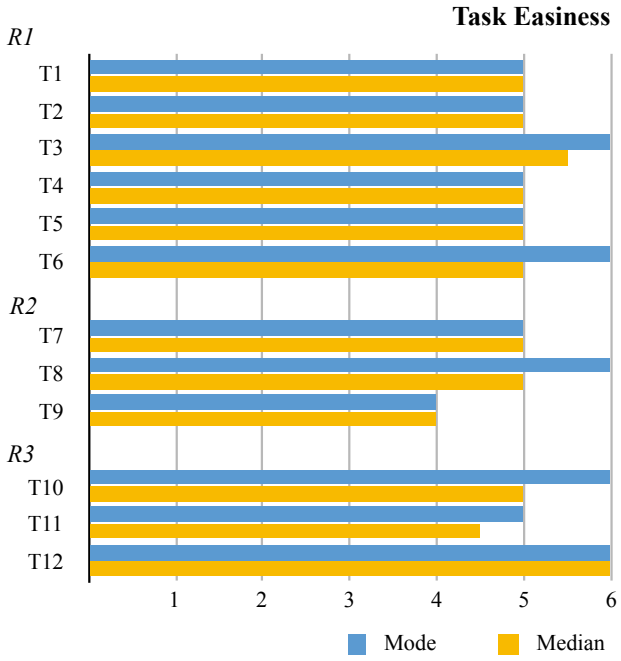

Figure I.6.3: Evaluation of the tasks easiness. *Question: The task was easy to complete (using the visualizations).* Higher score is better, (1 = Strongly disagree, 6 = Strongly agree).

the actual number of wrong responses, the overall feedback was that the tasks were easy to solve with the visualizations (*mode = 5, median = 5*). Friedman test does not reveal any statistical significance among the tasks of *R1* ($p = 0.27$). However, the two remaining groups have significant difference among their tasks. *R2* ($\chi^2 = 10.207, df = 2, p < .05$), *R3* ($\chi^2 = 6.5, df = 2, p < .05$). A post-hoc analysis using Conover's F test reveals that **T9** (*Find out the player who reached the closest score to your score.*) is statistically significantly difficult opposed to **T7** (*Characterize your score compared to other players.*) and **T8** (*Characterize your time spent by playing compared to other players.*). Likewise, there was a statistically significant difference in difficulty between **T11** (*Find out in which level you lost most points in the game.*) and **T12** (*Find out who reached the best score.*) in *R3*. Despite these observations, both the median and the mode of these two tasks is still greater or equal to 4 (= slightly agree). Therefore, we conclude that the hypothesis H2 was confirmed. The use of visualizations supports trainees' understanding and orientation in the game data.

H3 – Preferences in Using Visualizations

Our last hypothesis focuses on determining preferred visualizations for individual tasks. Figure I.6.4 presents the complete results obtained from a pair of questions focused on usefulness evaluation of visualizations with regards to the tasks. Although we find only

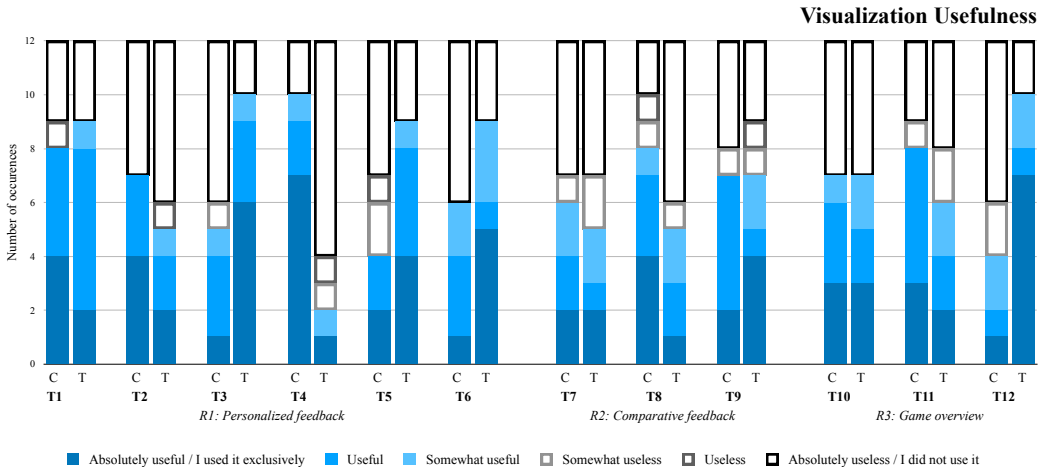

Figure I.6.4: Evaluation of the visualization usefulness with respect to the tasks. *Question: Evaluate the usefulness of the [*clustering — timeline*] visualization for the task.*; C = Clustering visualization, T = Timeline visualization.

slight differences between most of them, it does not apply for a subset of four tasks where we can observe stronger preference either for timeline (**T3, T5, T12**) or clustering (**T4**) visualization. To confirm our assumption on statistical significance, we performed one-tail Wilcoxon Matched Pairs Signed-rank Test. According to our expectations, there is no statistically significant preference for tasks **T1**, **T2**, **T6**–**T11**. Even **T5** is not significant ($W = 17, p = 0.28$). The three remaining tasks have significant preference for timeline visualization: **T3** (W = 8), **T12** (W = 8); and clustering visualization: **T4** (W = 7.5).

We confirm hypothesis H3 since there are at least four tasks, where we observed a statistically significant preference for either of the visualizations. However, only limited conclusions can be drawn for this hypothesis due to common tied values in our data. As a result, the sample size was often lowered by up to 5 samples (e. g., **T2, T7**). Also, the Wilcoxon test is known to be less sensitive when the sample size is very low ($N < 10$) and any difference that is statistically significant will have to be huge. Thus, further inspection with a larger sample is still needed.

I.7 Discussion

In this section, we summarize the results of the usability study, discuss limitations and lessons learned.

I.7.1 Summary of Evaluation Results

Design requirements are correct, and the tasks reflect user interests. The evaluation confirmed that the user requirements *R1–R3* distilled from interviews with four domain experts are meaningful and useful for trainees, and related tasks reflect user interests. Nevertheless, participants considered tasks of *R1* (personalized feedback) and *R3* (a brief overview of the overall game results and features) more meaningful when reflecting their gameplay than *R2* (comparative feedback).

Visualizations support trainees in the understanding of results. In general, trainees were able to complete given tasks correctly and, according to their response, tasks were easy to solve with the visual feedback. The evaluation revealed that some tasks were more difficult to solve than others. We aim to better support tasks identified as difficult in further research.

We did not find that any of the visualizations would better support any of the user requirements. On the other hand, we identified specific tasks for which one of the visualizations might be more appropriate. However, the results are uncertain due to data limitations. Further inspection with a larger sample is still needed to validate them.

We have received several suggestions for improvement. Trainees are not expected to interact with the feedback tool often. Therefore, it is important to reflect user experience in the design of the visualization tools so that they get familiar with them easily. We received particular suggestions and bug reports from the evaluation that we will reflect in the future development of the visual feedback. One participant noted that "the marking signifies the last change in score, but it's NOT the time I stopped playing". Three participants reported inconsistency in data presentation between the visualizations when the displayed time could vary by up to one second due to rounding off raw time-stamps. One participant suggested an improvement for the TIMELINE visualization: "There should be some buttons to select or deselect all players at once. Now the user has to click on each player individually."

I.7.2 Usability Study Limitations

Low number of participants. We decided for the field usability study since we wanted to reach as realistic settings as possible, which would be only hardly achievable in a controlled experiment. According to Rubin et al. [196], a truly experimental usability test achieving statistically valid results should be conducted with a minimum of 10 to 12 participants per condition. And although the sample size of 12 participants is not unusual in similar studies in general, we are aware of this weakness for claiming a strong confirmation of our findings. Since research has shown that a sample size of 4 to 5 participants can expose about 80 % of usability issues [196], and since the hypotheses were confirmed, and initial outcomes are positive, we consider the results as promising and entitling us to elaborate the feedback

visualizations further.

Dataset limitations. The dataset used in the evaluation was not synthetic. Thus some of the tasks defined prior to the experiment do not have clear "one-value only" answers. There were multiple correct solutions or data was unclear. As a result, some of the tasks were more difficult to solve, or the solution was not straightforward (e.g., **T9**, **T11**). We also faced three technical issues on the cyber range infrastructure when some of the game events were not recorded. As a result, the visualizations did not reflect the real-world situation of three participants, which was confusing for them and probably affected some of the responses in **T8**.

Ambiguous responses to visualization preference. We identified eight ambiguous responses where, despite the instructions, participants tick *Absolutely useful / I used it exclusively* in one visualization and different option than *Absolutely useless / I did not use it* in the other. Three of the responses were from the same participant. Some of the participants also noted that they use the table (which is, in fact, a part of the TIMELINE) instead of visualization itself. Since we plan to repeat the usability study, we are going to revise the way of presenting the questions to reduce the ambiguity and extract more qualitative information on the actual use of visualizations.

I.7.3 Observations and Lessons Learned

Trainees prefer exploration of personal results to the overall game results and comparison with others. Detailed inspection of the results revealed that the participants were primarily interested in their score (tasks of *R1*), followed by the overall awareness of the game (tasks of *R3*). We assume that the primary objective of a player is to get insight into his/her gameplay (a score development, the time they spent playing the game and its parts). Further, they are interested in the overall game situation without bothering too much with a detailed comparison with others (tasks of *R2*). We assume that the comparative perspective is meaningful only in specific cases – e.g. when two friends want to compare their scores. Our initial findings and confirmation of H1 open a new research topic of determining the essential information for trainees and their additional support in post-game feedback visualizations. Since this is far beyond the scope of this paper, we leave it for our follow-up work.

Easy to decode design is not mandatory. Our preliminary expectations that a much simpler CLUSTERING visualization would be more useful and used than more complex TIMELINE visualization have not been confirmed. Even though both visualizations were designed for a different type of tasks, there is considerable overlap in their capabilities in answering the same questions. For example, both of them offer a straightforward way of finding the final score, and the scores reached in individual levels. However, the study did not reveal preference of the CLUSTERING to TIMELINE. Data of only four of the tasks reported the preference for either of them. Although the intuitiveness of the design is still mandatory due to the restrictions of the reflection phase, the easy to decode design seems

not to be essential, as we expected. The users are willing to use a more complex variant if they provide more details. There were also a participant who prefers simple presentation in tabular form from fancy visualizations.

Trainees tend to perceive time subjectively. The task **T8** (*Characterize your time spent by playing compared to other players.*) was ranked with five options from *I was one of the slowest players* to *I was one of the fastest players.* Answers to this question included the most number of wrong responses (58%). Participants either under- or overestimated their finish time compared to others. Unfortunately, none of them explicitly commented on why. We assume that participants reflected their impression from real-world time (someones started playing a little earlier, some later) rather than precise relative game time represented in visualizations.

Do not mix time spans with a different meaning, even if they are in separate visualizations and explained by a legend. We noted yet another confusion with the time-related tasks. Whereas CLUSTERING visualization shows average and total time based on the trainees' data, TIMELINE visualization displays estimated time for each level (colored diagonal stripes on the background in Figure I.0.1). **T10** (*Find out how much time was assigned for playing the game.*) targets the latter one. All the participants who made a mistake indicated the maximal time from the CLUSTERING visualization instead.

Users can interpret game results subjectively. Game results consist of a score and time in which the score was reached. In general, the best players are considered those who achieved the best score. Nevertheless, also trainees who did not get the absolute best score but solved the task quickly can be considered as very successful. The task **T11** (*Is there somebody who reached a high score in significantly short time? If so, insert his Player ID.*) was to reveal such trainees. However, the task has a considerably high number of wrong responses (50%). From the detailed inspection of the results, we found out that the biggest issue is unclear data that makes the task difficult. There was no single recognizable answer in the data. We (the authors) agreed that the correct response is the player who finished as the first one. He was also one of the two players with the highest score. Most of the participants (in five cases) marked the same player. One of the participants marked both players with the highest score (i.e., partially correct response). The rest of the participants marked different ones, usually those who finished earlier but had a rather low score. A solution to this problem could be to include data storytelling principles so that users immediately see what is important by providing a "narrative", as discussed in [74], for instance.

The clustering approach can be confusing in specific cases Responses of **T9** (*Find out the player who reached the closest score to your score.*) revealed one weakness in the design of the CLUSTERING visualization. One participant, who used this visualization solely and provided the wrong player ID, inspected only the closest neighborhood where he found the wrong answer. The correct one was placed too far right from his position since this participant was one of the slowest, as illustrated in Figure I.7.1. In this situation users tend to prefer closer neighbours to more distant points. A solution to this problem could be

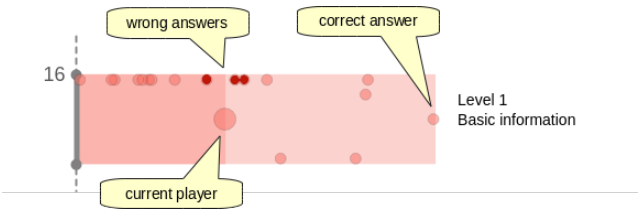

Figure I.7.1: Clustering weakness – the users tend to select closer neighbors while omitting the distant ones.

to gradually highlight those who are in, e.g., 10-25-50 % dispersion around both horizontal and vertical dimensions. Additionally, the pop-up tooltip (raised when the mouse cursor hovers over the dot) could be extended with Top 3 better and Top 3 worse trainees regarding the observed one.

I.8 Conclusion and Future Work

The work we presented in this paper focuses on improving post-game feedback for players (trainees) of serious multi-level cybersecurity games by using interactive visualizations. The feedback is one of the critical phases of the learning process. However, the way of presenting the results is often limited to plain scoreboards presenting only the total scores and generalized feedback on the most common issues observed during the gameplay. We improve the user experience through design and implementation of two interactive visualizations. The visualizations improve overall situational awareness and insight into the gameplay. Also, they provide a straightforward way for comparison of individual trainees. A demo version with test data is available online at `http://kypo-summer.surge.sh/`.

We collaborated with four cybersecurity education experts on defining visualization requirements. Together, we formulated a set of tasks that address three areas of trainees' interest: gameplay overview, person-centered feedback, and comparative feedback. Through the usability study held with participants of Summer School in Cybersecurity, we evaluated three hypotheses related to (a) meaningfulness of the user requirements and related tasks, (b) usefulness, and (c) preference of the visualizations regarding the tasks. All three hypotheses were confirmed, but due to a low sample size (only 12 participants), some of the conclusions will be addressed in our future work. Usability study also confirmed the practical usefulness of the visualizations and pointed out several topics worth further investigation. Namely, defining the set of preferred information for trainees (such as correct or incorrect attempts) and their additional support in feedback visualizations. Further, we collected valuable feedback on the strengths and weaknesses of the visualizations, which we want to address in the implementation. Although the results are inconclusive, the usability study also revealed a preference for one of the visualizations concerning the specific tasks.

Our work is still in progress. In the future, we intend to further improve the previously mentioned following aspects. Our overall goal is to provide, through interactive visualizations, personalized feedback to increase gained knowledge. Our ongoing work on integrating both visualizations into the cyber range web portal is just the first step. The integration also implies interconnecting and extending the interaction capabilities of both visualizations. For example, selecting a trainee in one visualization will highlight the same relevant data in the other. We also want to integrate the visualizations into the gameplay and extend their capabilities toward personalized post-level feedback. The aim is to provide an instant user-specific overview of strengths and weaknesses based on the user's actions. Our research will include experimental evaluation to confirm and strengthen the initial conclusions from the presented usability study as well as evaluation of new features. Hands-on cybersecurity training is a part of regular university lectures. Therefore, we prefer running more extensive field study over standard laboratory evaluation on synthetic data.

The visualizations were designed to provide timely feedback to trainees. However, there are other users involved in the CTF life cycle that would benefit from an interactive exploration of CTF results. The visualizations discussed in this paper enable *supervisors* (the educators who oversee the game) to reflect the overall results so that they can assess their interventions during the game sessions. Similarly, *game designers* (the authors of the content) can utilize the visualizations in their workflow to evaluate game parameters. We will address these issues in future research as well.

Acknowledgment

The authors would like to thank the cybersecurity experts for providing the feedback and invaluable insight into the target domain. This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822). Computational resources were provided by the European Regional Development Fund Project CERIT Scientific Cloud (No. CZ.02.1.01/0.0/0.0/16_013/0001802).

Article J

Timely Feedback in Unstructured Cybersecurity Exercises

Jan Vykopal[1], Radek Ošlejšek[2], Karolína Burská[2], Kristína Zákopčanová[2]

[1] Masaryk University, Institute of Computer Science, Brno, Czech Republic
[2] Masaryk University, Faculty of Informatics, Brno, Czech Republic

SIGCSE – Proceedings of Special Interest Group on Computer Science Education. ACM, 2018, p. 173-178, 6 pp.

Abstract

Cyber defence exercises are intensive, hands-on learning events for teams of professionals who gain or develop their skills to successfully prevent and respond to cyber attacks. The exercises mimic the real-life, routine operation of an organization which is being attacked by an unknown offender. Teams of learners receive very limited immediate feedback from the instructors during the exercise; they can usually see only a scoreboard showing the aggregated gain or loss of points for particular tasks. An in-depth analysis of learners' actions requires considerable human effort, which results in days or weeks of delay. The intensive experience is thus not followed by proper feedback facilitating actual learning, and this diminishes the effect of the exercise.

In this initial work, we investigate how to provide valuable feedback to learners right after the exercise without any unnecessary delay. Based on the scoring system of a cyber defence exercise, we have developed a new feedback tool that presents an interactive, personalized timeline of exercise events. We deployed this tool during an international exercise, where we monitored participants' interactions and gathered their reflections. The results show that learners did use the new tool and rated it positively. Since this new feature is not bound to a particular defence exercise, it can be applied to all exercises that employ scoring based

on the evaluation of individual exercise objectives. As a result, it enables the learner to immediately reflect on the experience gained.

J.1 Introduction

Cyber attacks threatening ICT infrastructure have become routine. Their intensity and complexity are growing with the increasing number of interconnected devices exposed to attackers and the influx of new vulnerabilities being revealed each year. Unfortunately, there is a significant global shortage of cybersecurity workers equipped with the skills necessary for preventing or responding to the attacks [223].

Cyber defence exercises (CDX) [71] represent a popular type of training that aims to fill this skill gap. They enable participants to experience cyber attacks first-hand with real-life limitations, including a lack of information and resources, the need for communication and making decisions under stress.

CDX are usually intensive, short-term events lasting several days. Tens to hundreds of professional learners participate and are grouped in teams. The target groups are administrators of ICT systems, incident responders, and security managers. The exercises deliver rich immediate experience of ongoing attacks and the opportunity to practice crisis procedures and techniques.

In contrast to structured, step-by-step hands-on training guided by an instructor, teams of learners have to figure out all the issues on their own, in the order they agree on, within the team. These settings simulate real operation but prevent any direct feedback from instructors (exercise organizers). Learners can only presume what they did was correct, what worked and what did not. The only feedback they are given during the exercise is often through an exercise score with no further details about score breakdown. Some exercise organizers provide technical reports after the exercise, which reveal some details highlighting important moments from the perspective of a particular team of learners. The after-action report is sometimes also complemented by a short workshop, which provides an opportunity to discuss the content of the exercise with the instructors in person. Nevertheless, all these methods of feedback are delivered with a significant delay after the actual exercise because they require preparation from the instructors that cannot begin before the end of the exercise.

In this paper, we study whether learners benefit from simple, but individualized feedback provided just after the end of a two-day intensive exercise. In our study, each team was provided with an interactive timeline of its score development during the exercise, with important events emphasized. The timeline was generated automatically from data stored by an existing scoring system. All interactions of exercise participants (mouse clicks and movements) were logged with the scoring timeline. After that, participants were asked to fill out short evaluation questionnaire. The data and answers we obtained show that learners valued the feedback, even though they still lack more details about particular events.

J.2 State of the art

Research on providing feedback in complex and unstructured cybersecurity exercises is very scarce. The following overview is therefore based not only on a review of academic literature but also on technical reports published by organizers and the experience of authors who participated in several CDXs.

One of the world's largest exercise is Locked Shields [170], which is organized annually by the NATO Cooperative Cyber Defence Centre of Excellence in Tallinn, Estonia. Immediate brief feedback to learners is provided at a so-called "hot wash-up" session right after the exercise since any time lag will diminish the learning impact [146]. Educators provide a summary of the exercise's progression and comment on key moments that drove it. More comprehensive and detailed feedback is available only at a workshop which is held a month later, and furthermore, not all learners come to this event.

Another international exercise organized by NATO is Cyber Coalition. Very brief feedback is provided a day later in-person at a hot wash-up session. More specific information is available only in the form of an after-action report at a workshop, which takes place a month later [171]. A very similar type of feedback, with an even longer delay is provided in Cyber Europe, another international exercise organized by ENISA. [80]

Granåsen and Andersson [95] focused on measuring team effectiveness in CDXs. They thoroughly analysed system logs, observer reports, and surveys collected during Baltic cyber shield 2010, a multi-national civil-military CDX. They concluded that these multiple data sources provided valuable insight into the exercise's run. However, they did not mention how to use these data for providing feedback. The only feedback provided to learners was during a virtual after-action review the day after the exercise, where only the leaders of learners' and organizers' teams summarized and discussed their experience.

Henshel et al. [108] also focused on the assessment model and metrics for team proficiency in CDXs. They analysed learners' and observers' input from surveys and intrusion detection logs from the Cyber Shield 2015 exercise. Similarly to the Baltic cyber shield exercise, the feedback provided to learners was not based on an analysis of acquired data since the analysis was done manually and required significant human effort.

Since existing CDXs do not provide any timely and personalized feedback to the learners, we also mention a feedback-related study in learning, which does not concern CDXs. Gibbs and Taylor [91] focused on the theory that personalized feedback does not hold such importance or a value compared to its time-consumption for the instructor.

Table J.3.1: Phases of the exercise with time allocation

Order	Phase	Duration	Day
1	Exercise familirization	3 hrs	1
2	Actual exercise	6 hrs	2
3	Post-exercise survey	5 mins	2
4	Break	25 mins	2
5	Scoring timeline interaction	10 mins	2
6	Scoring timeline survey	5 mins	2
7	Quick exercise debriefing	15 mins	2

J.3 Experiment setup

In this experiment, we studied the behaviour and interactions of participants at a complex cyber defence exercise held on May 23–24, 2017 at Masaryk University, Brno, Czech Republic. The exercise is focused on defending critical information infrastructure (particularly railway infrastructure administration) against skilled and coordinated attackers.

First, the learners got access to the exercise infrastructure to become familiar with virtual hosts in their network. Then, they took part in an intensive exercise where they faced challenges posed by simulated attackers and legitimate users. Right after the end of the exercise, the learners were asked to express their immediate impressions about the exercise in a short post-exercise survey. After a short break, a timeline depicting the score of each team was presented in the exercise portal. Finally, the learners were asked to evaluate the timeline via a very short questionnaire. Time allocated for each phase of the experiment is shown in Table J.3.1.

J.3.1 Exercise participants

The experiment involved a Red vs. Blue exercise with 40 participants working in an emulated ICT infrastructure. The structure of the exercise is inspired by the Locked Shield exercise [170, 244]. The participants were divided into four groups according to their role and tasks in the exercise. Their interactions are depicted in Figure J.3.1.

Twenty professional learners formed five *Blue teams* (T1–T5) which were put into the role of emergency security teams sent into five organizations to recover compromised networks. Each team of 4 learners was responsible for securing the compromised networks, dealing with the attacks, collaborating with other emergency teams, and collaborating with the coordinator of the operation and media representatives. They had to follow the exercise's rules and local cybersecurity law. Each team represented one real cybersecurity response team from one country in Central Europe.

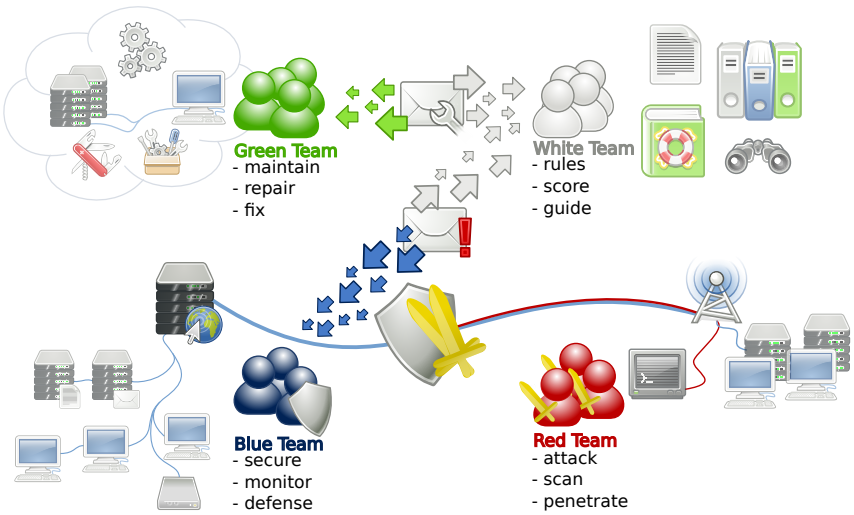

Figure J.3.1: Exercise participants, their interactions and tasks.

All attacks against infrastructure defended by Blue teams were conducted by *Red team*. This team consisted of cyber security professionals who carefully followed a predefined attack scenario to equally load the Blue teams. This means they should not use any other arbitrary means of attack against the Blue teams. Based on the success of attacks, the Red team assigns penalty points to the Blue teams since the amount of points is based on non-trivial factors that need expert review.

Exercise managers, referees, organizers, and instructors worked in *White team*. During the exercise, this team assigns tasks (called injects) to the Blue teams and thus simulates the requests of many entities, such as legitimate users of the defended organization, the operation coordinator which needs situational reports, media inquiries, and law enforcement agencies. Then the White team assesses the promptness and quality of a Blue teams's reactions to these tasks and assigns penalties and points.

Finally, the *Green team* is a group of operators and system administrators responsible for the exercise infrastructure. They have full access to the exercise network so they can provide assistance to Blue teams in trouble in an exchange for penalty points.

J.3.2 Exercise phases

Before the actual exercise (Phase 1 in Table J.3.1), learners are provided with a background story to introduce them to the situation before they enter the compromised networks. Then they access their part of the emulated network for 3 hours to get familiar with the exercise infrastructure. This is very important since the exercise is not set in a known environment and learners have no previous knowledge about who is who in the fictitious scenario (e. g., users in their organization, a popular news portal, superordinate security team).

The exercise (Phase 2) is driven by a detailed scenario which includes the actions of attackers (Red team) and assignments for the defenders prepared by the organizers (White team). Attackers exploit specific vulnerabilities left in the compromised network in a fixed order. This follows a common attack life cycle in a critical information infrastructure. On top of that, learners should also answer media inquires and requests from users doing their routine job in the defended network. The performance of each Blue team is scored based on successful attacks or their mitigation, the availability of specified critical services and the quality of reporting and communication. The score is either computed automatically from events, processed by the logging infrastructure (e. g., a penalty for inaccessible services) or entered manually (e. g., attacks completed by the Red team). An aggregated score is shown to participants in real-time. Table J.3.2 shows the structure of the scoreboard and the values of aggregated score.

Table J.3.2: The scoreboard presented to the learners during the exercise.

Team	Services	Attacks	Injects	Users	Access	Total
T1	91,843	-8,500	9,000	-1,100	0	91,243
T5	92,230	-5,000	3,600	-400	0	90,430
T2	81,280	-10,750	6,425	-4,000	0	72,955
T4	74,518	-11,000	6,650	0	-4,000	66,168
T3	85,756	-12,000	2,475	-1,700	-9,500	65,031

Note: Teams are sorted according to their final score.
"Injects" is an abbreviation for communication injects of the White team.

Immediately after the end of the exercise, we asked the learners to evaluate the exercise and their experience by rating several statements using the Likert scale (Phase 3).

J.3.3 Scoring timeline application

After a break (Phase 4), the score acquired by each team during the exercise was presented to the Blue teams in the form of an interactive application, as shown in Figure J.3.2 (Phase 5). It provided automatically generated personalized feedback. Members of each team could only see their own scoring timeline with individual exercise events.

The initial score of each team is 100 000 points. In the graph, the main, predominantly descending line represents the development of a team's total score over time. The score is computed from penalties and awarded points that were either recorded automatically for the inaccessibility of required network services or assigned manually by the Red or White team. The colourful dots are interactive and they are related just to the manual rating. The red dots represent Red team penalties, white dots represent the rating of communication injects by the White team, yellow dots indicate the rating of user simulated injects by the White team, and grey dots indicate requests for assistance from the Green team (to grant

temporary remote access to a machine or revert to the initial state). Each dot contains textual information that specifies the reason for the rating. This information is shown in each dot's tooltip.

Learners were able to provide us with their reflection on their penalty and awarded points very easily by clicking on the coloured dots and choosing one of predefined options (Phase 5; see right-hand side of Figure 2), e. g., whether they recognized the attack or not, or why they did not respond to the inject of the White team. Moreover, all scoring timeline interactions, including mouse clicks, mouse movements, and selected options were logged. This data, together with answers from a short survey on the scoring timeline (Phase 6) were used to evaluate the usefulness of the timely feedback.

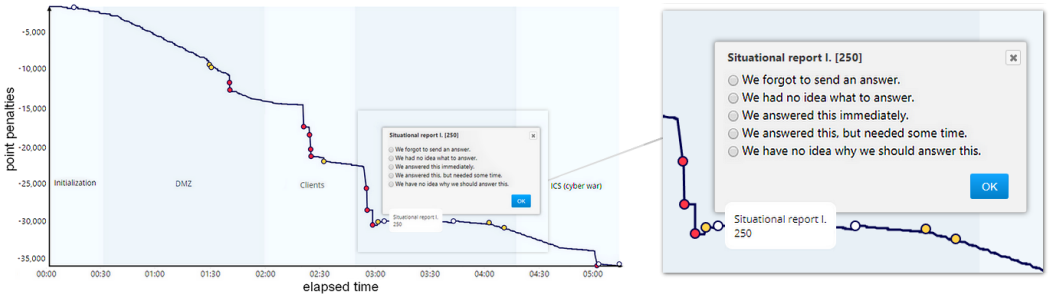

Figure J.3.2: A screenshot of a scoring timeline providing personalized feedback for each team right after the exercise.

Finally, representatives of the Red and White team provided a short debrief of the exercise ("hot wash-up", Phase 7) from their perspective. They highlighted important breaking points of exercise and pointed out exemplary or interesting decisions and actions took by Blue teams. This part is mentioned here only for completeness, no input from learners was required for the experiment.

J.4 Results

J.4.1 Post-exercise survey

The post-exercise survey was focused on general qualitative aspects of the exercise. Relevant statements to this study are listed in Table J.4.1 and marked as E1–E4 for further reference. The statistical distribution of individual answers across all teams is depicted in Figure J.4.1.

We collected answers from all 20 participants. However, four of them did not provide their identification and so their answers were omitted from the team statistics. Team values were computed as the average from a five-point Likert scale answers (1 = strongly disagree, 5 = completely agree). The *Average* column represents the average value across all answers regardless of teams.

Table J.4.1: Team statistics for the post-exercise and scoring timeline surveys

	Statement	Team2	Team1	Team5	Team4	Team3	Avg
E1:	My knowledge and skills were sufficient.	3.75	3	2.6	2.5	2	3.05
E2:	I found exercise difficult for me.	3.33	3	4	4.25	5	3.8
E3:	Exercise was well organized and structured.	2.66	3.25	3.3	4	5	3.75
E4:	Exercise was beneficial and useful to me.	2.66	3.5	4	4.5	5	3.85
F1:	The scoring timeline of my team displayed after the end of the exercise provided useful feedback.	3.25	2.5	-	4.25	3.66	3.53
F2:	Do you have any comments on the scoring timeline?	D	M	-	M	M	

1 = strongly disagree, 5 = completely agree, D = there was a delay in inserting points by a Red and White team, M = add more details about the depicted events

The aim of E1 and E2 was to reveal the level of expertise of individual teams and the difficulty of the exercise. Individual answers to E1 significantly varied, which indicates that learners had significantly different expertise. However, the average values calculated for each team reveal that the exercise was well balanced with no extremely weak or strong team. The answers to E2 indicate that the overall difficulty of the exercise was considered as rather high.

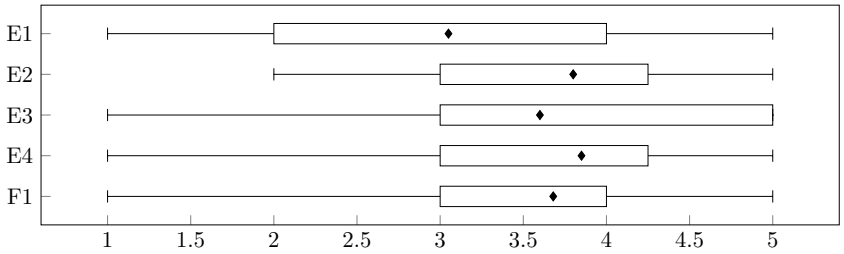

Figure J.4.1: Distribution of all answers to E1 – E4 and F1.

Two statements in the questionnaire, E3 and E4, were focused on satisfaction with the organizational aspects and usefulness of the exercise. Both statements brought very similar answers. Teams that were more satisfied with the organization also considered the exercise more beneficial. Even though the opinion differed across the teams, learners considered the exercise rather beneficial and well organized in general.

J.4.2 Scoring timeline interaction

In order to evaluate the scoring timeline, we were actively recording learners' interactions with a tool. We obtained data from 18 learners, 2 learners were missing due to technical issues. The data consisted of 2,994 individual low-level events (mouse clicks, mouse hovers, etc.). Moreover, we recorded heatmaps of mouse positions on the screen (see an example

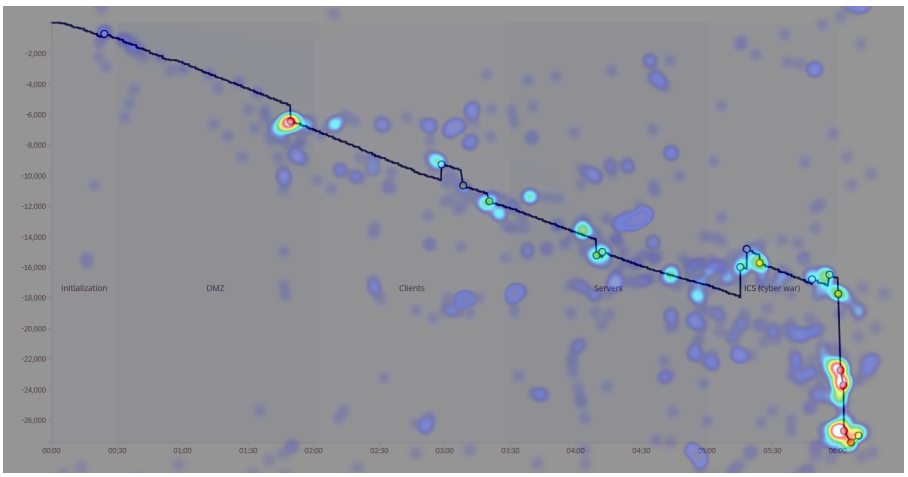

Figure J.4.2: An example of a heatmap of mouse movements and clicks on a screen of a scoring timeline.

in Figure J.4.2). Putting this data together, we were able to estimate the focus of the learners during their exploration of the scoring timeline. A deeper analysis did not reveal any preference patterns in the sense that learners would be more interested in some kind of objectives, such as Red team attacks, White team injects, or penalties from later critical phases of the exercise. On the contrary, it seems that the learners were interested in all the penalties and awards roughly the same.

Analyses of timestamps revealed that the time spent by individual learners with the scoring timeline ranged between 1m 22s and 8m 27s It is worth noting that there was no significant difference between teams. They spent approximately 3 to 5 minutes with the feedback application on average. We also did not find any relation between the time spent with the timeline and the willingness to provide their reflection on particular penalties or awards. Many learners just spent a long time with only a passive exploration of the timeline.

Seven learners from four teams also gave us an active reflection to penalties in addition to passively exploring the scoring timeline. The values in Table J.4.2 represent the numbers of collected answers per team and objective type. Blue team 1 is omitted from the table because we got no data from them. These results are discussed in Section J.5.

J.4.3 Scoring timeline survey

The usefulness of the feedback provided via the scoring timeline was evaluated with a short survey. We got answers from 13 learners (out of 20) because Blue team 5 did not respond at all. The statements and their team statistics are shown in Table J.4.1 under the labels F1 and F2.

Table J.4.2: Numbers of responses of each team to objectives.

	Teams				
Objectives	T2	T3	T4	T5	$\sum$
Red team attacks	7	13	5	1	26
Users injects	5	7	0	1	13
Communication injects	0	5	0	2	7
Green team assistance	0	4	2	0	6

The data shows that the usefulness of the feedback can be considered to be "rather useful" (the average value across all the teams is 3.53). A detailed distribution of answers depicted in Figure J.4.3 reveals that the most successful team considered the feedback less useful than other teams.

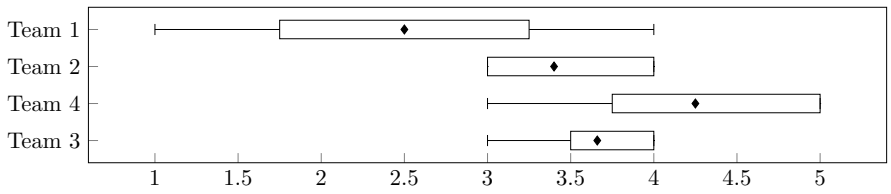

Figure J.4.3: Distribution of answers to F1. Teams are sorted according to the final score (the best on the top).

Answers to the open question F2 provided four comments on possible improvements. Three teams requested more details about penalties. They would appreciate knowing "how it happened" in addition to "what happened" (marked as *M* in Table J.4.1). One comment objected that "some attacks were happening way sooner in reality than on the timeline" (marked as *D* in Table J.4.1). This is true because attack penalties were inserted by the Red team manually with some delay.

J.5 Discussion

The validity of the timely feedback evaluation would be affected by dissatisfaction with the exercise so the learners would not be interested in the feedback at all or they would provide distorted data. Dissatisfaction can be caused by poor organization, unfulfilled expectations, or by a disparity between the difficulty of the exercise and the knowledge of learners. In general, this exercise was considered slightly difficult and was also assessed positively (see Section J.4.1). Although the exercise was attended by skilled teams whose members had previous experience with similar exercises, they still considered it rather difficult, challenging and useful. Therefore, we believe that conclusions drawn from the analysis based on this particular exercise and its participants are plausible. In the following part, we put together

the results of the post-exercise survey, the scoring timeline survey, and the exploration of the timeline and analyse their mutual relationships.

Teams sought out feedback A deeper analysis of learners' interactions with the scoring timeline shows that all teams were using it intensively, regardless of what reflection they provided in the scoring timeline survey. All teams explored all the penalties depicted in their timeline. They also gave us an evaluation of the majority of displayed objectives. The only exception was Team 4, probably due to technical issues. Team 5, which did not respond to the scoring timeline survey at all, still explored their timeline actively. Team 1 rated the scoring timeline as less useful than other teams. It was the most successful team according to the final score and, therefore, the feedback may not have been so interesting for them because they might have already known about their failures. However, even this team was interested in the timely feedback because they explored the scoring timeline for the longest.

A need for more detail Answers to the open question F2 confirmed our assumption that more precise and more detailed information provided by the timely feedback makes the feedback even more attractive to learners. Nevertheless, the need for more detail can also be indirectly inferred. For example, a further analysis of the White team's injects (users and communication injects) indicates that teams often underestimated time-dependant response to this type of "soft" request. Teams could either consider these requests "annoying" and not so important in comparison with attacks or they could just be too busy with the attacks, for instance. A deeper understanding of their behaviour also requires collecting more detailed and better structured data from the timely feedback.

Benefits for instructors Although the scoring feedback was intended primarily for learners, the previous discussion shows that it is very valuable also for organizers and educators who can learn about the exercise and then fine tune its parameters, e. g., by better scheduling of the White team's injects with respect to the attacks. Since learners are not usually aware of this value, they do not make more effort than necessary into providing quality and valuable information for instructors. However, if providing reflections is intuitive and quick for them, they are motivated to use it. Furthermore, since the feedback is generated automatically, organizers can get valuable data without any additional effort.

Limitations of the study This study is limited in two respects. First, data were obtained from a single exercise with a relatively small group of participants. The reason is that the organization of such a complex exercise is expensive and time-consuming and it is organized rarely for only a narrow group of experts. Nevertheless, we consider the sample to be representative because the learners were highly qualified experts, often with experience in similar exercises and the exercise received a positive rating from them. Second, the timeline evaluation questionnaire was simple and freely structured. However, this was intentional

since the survey took place at the end of an exhausting two-day exercise and thus a more sophisticated questionnaire might not have guaranteed more precise results.

J.6 Conclusions and Future work

To best of our knowledge, this paper is the first attempt to study the means of providing feedback to learners participating in cyber defence exercises. The literature review and our own experience showed that feedback provided in state-of-the-art exercises is very limited or delayed. The most often used method is an exercise scoreboard displayed to the learners throughout the exercise. Another method is a short verbal evaluation by exercise observers or organizers right after the end the exercise or its phases. The last commonly used method is after-action reports highlighting key conclusions from a laborious manual analysis of heterogeneous data acquired during the exercise (survey, written communication, scoring and monitoring logs, and checks).

The lack of timely feedback results in the learners having limited opportunity to learn from their experience. They undergo numerous real-life situations during the exercise, but they are not supplied with an explanation why such situations occur. We therefore complemented these means by a novel approach which provides feedback to learners right after the end of the exercise with no additional effort required from educators. Learners can explore a scoring timeline depicting increases and decreases in their team's score and display details about individual events (name and type of the exercise objective, and the number of awarded points or penalty points). Also, they have an opportunity to provide their reflections to educators and indicate their awareness about a particular objective and its solution. The exploration and interaction with the scoring timeline enables learners to reflect on their experience and thus strengthen the learning impact of the exercise.

In order to evaluate this approach, we ran an experiment involving a two-day, complex cyber defence exercise with 24 objectives, and 20 professional learners from five security teams. The results, based on an analysis of user surveys and interactions with the new tool, suggest that learners welcomed the new feature even though the feedback was mined automatically and thus provided a very limited level of detail about particular events.

The experiment also outlined directions for future work. First, learners would appreciate more detail about a particular event in the timeline. This can be easily done by adding a detailed description of the objective related to the event. Another option is to extend the scoring application so that instructors can not only assign points, but also provide a comment on each exercise objective. Second, the timeline could be enriched with a display of the relationship between all exercise objectives. This would highlight that an event that the team may not have been aware of was caused by several previous events they encountered. The context can be then built not only from a time perspective, but also from the topology of the exercise network. Providing information on which host or service was affected by

the particular event may help learners to recall the particular situation and understand it better. Finally, the ultimate goal is to provide a "replay function", which would show how attackers proceeded and what could have been done better to prevent or mitigate attacks.

Acknowledgments

This research was supported by the Security Research Programme of the Czech Republic 2015-2020 (BV III/1–VS) granted by the Ministry of the Interior of the Czech Republicunder No. VI20162019014 – Simulation, detection, and mitigation of cyber threats endangering critical infrastructure.

Article K

Evaluation of Cyber Defense Exercises Using Visual Analytics Process

Radek Ošlejšek[1], Jan Vykopal[2], Karolína Burská[1], Vít Rusňák[2]

[1] Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2] Masaryk University, Institute of Computer Science, Brno, Czech Republic

FIE – IEEE Frontiers in Education Conference. IEEE, 2018, 9 pp.

Abstract

This Innovative Practice Full Paper addresses modern cyber ranges which represent unified platforms that offer efficient organization of complex hands-on exercises where participants can train their cybersecurity skills. However, the functionality targets mostly learners who are the primary users. Support of organizers performing analytic and evaluation tasks is weak and ad-hoc. It makes harder to improve the quality of an exercise, particularly its impact on learners. In this paper, we present an application of a well-structured visual analytics process to the organization of cyber exercises. We illustrate that the classification derived from the adoption of the visual analytics process helps to clarify and formalize analytical tasks of educators and enables their systematic support in cyber ranges. We demonstrate an application of our approach on a particular series of eight exercises we have organized in last three years. We believe the presented approach is beneficial for anyone involved in preparation and execution of any complex exercise.

K.1 Introduction

Visual analytics (VA) is the science of analytical reasoning supported by interactive visual interfaces [253]. It is applied in various fields from biology or weather forecast [132, 129, 137, 64, 65] to education [232, 230]. As a specific case, we can consider applied cybersecurity training which is of our focus.

Various hands-on training programs which aim at improving attacking or defending skills of learners often augment theoretical cybersecurity education. While Capture the Flag (CTF) games focus on the attacking skills and learners solve one task at a time, Cyber Defense Exercises (CDX) are more complex events [71]. They mimic real-world operations of an organization under the attack of an unknown offender, and their participants work in teams on several issues at a time.

CDXs usually run in virtual environments called cyber ranges [2, 243]. Cyber ranges provide access to virtual computer networks where learners exercise their skills and abilities to protect the infrastructure against the attackers. The development in cyber ranges focuses mostly on tooling for learners who are the primary users and the instant assessment. Less attention is paid to analytical tools for organizers, which makes an in-depth evaluation and analysis tasks laborious and time-consuming. Nevertheless, these tasks are crucial in the process of continuous improvement of CDX events.

Related analytical tasks and visualizations discussed in this paper clarify analytical interests of organizers and provide a mapping to the general visual analytics model. This systematization is crucial for building awareness of organizational aspects so that the organizational process can be automated in cyber ranges. To demonstrate practical applicability, we present experience gained from the organization of a particular CDX series.

In the rest of this section, we describe key features of principles of visual analytics framework and cyber defense exercises. Section K.2 discusses an adaptation and interpretation of the visual analytics framework in the context of CDX organization and its evaluation. We demonstrate a practical application of our approach on a case study described in Section K.3. Lessons learned from our experience follow in Section K.4. Section K.5 concludes the paper with the outline of follow-up work and research opportunities.

K.1.1 Visual Analytics

Keim et al. proposed a formal description of the visual analytics (VA) process in [124, 123]. They defined basic terms like data, models, visualization, and knowledge together with their modeling and analytical processes. However, this model is primarily system-driven. It focuses on automated data analyses and does not consider details of user-driven analytical tasks forming the knowledge via human reasoning.

Sacha et al. in [199] provide a solution that extends the computer part of Keim's model

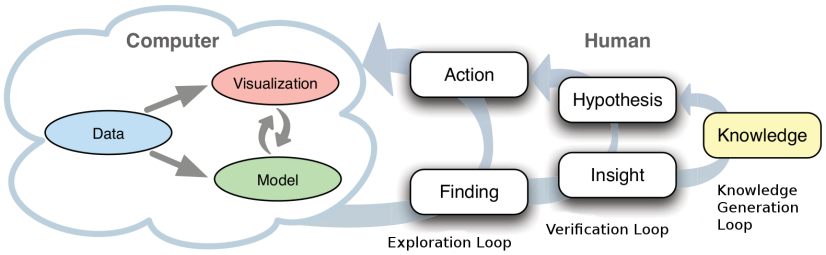

Figure K.1.1: An overview of the knowledge generation model for visual analytics [199].

with hierarchically connected human loops, as shown in Figure K.1.1. In this model, we define *knowledge* as a "justified belief" that our understandings in a problem domain are correct [25]. A central role in building knowledge play *hypotheses*. In the beginning, they can be vaguely defined using many unknown factors; then they can be gradually refined to produce deeper *insight* into a problem domain. Sufficiently approved insights and hypotheses can be accepted as new knowledge which can affect or initialize further hypotheses. During the *exploration loop*, analysts either verify or disprove hypotheses via *actions* that manipulate data and models utilizing interactive visualizations. Gained *findings* have no interpretation. A finding would be an unusual peak in a graph, for instance, that attracts analyst's attention. To understand the peak and then to gain insight, the peak has to be interpreted by the analyst. It often requires further actions to be performed. Meaningful findings can lead to gaining insight into the problem domain.

K.1.2 Cyber Defense Exercises

The CDX is an exhausting event which usually spans one to several days of a very intensive engagement of learners. It includes familiarization with the infrastructure, hands-on experience, and the evaluation phases. Hands-on part runs by a prescribed game scenario. However, the whole CDX life cycle is even more demanding for organizers. It spans several months and involves dozens of highly skilled people in multiple domains (cybersecurity, education, law).

Persons involved in the CDX usually form four teams. Learners, mostly ICT professionals, are organized in several Blue teams consisting of at least three people. *Red team* members represent attackers who run attacks against the Blue teams. Scoring and controlling game scenario and rules are tasks of the *White team* members. They also represent several avatar characters (company users, management, lawyers), or journalists who interrupt the game with various inquiries on the Blue teams. Members of the *Green team* maintain the underlying infrastructure of the exercise.

As we can deduce, CDX examines both technical and soft skills of learners. Every Blue team tries to protect a dedicated IT infrastructure while facing multiple issues at a time.

They are forced to prioritize and assign tasks ranging from technical issues (e. g., hacked server) to interaction with avatar characters (e. g., creating press news). The learners can only presume whether their actions were correct or not based on the minimal feedback in the form of a total score of the team. Example techniques for automated assessment of learners performance are in [67, 6]. The organizers can usually access detailed overview based on game scoring rules.

The actual exercise (gameplay) is only one of the four phases of the CDX life-cycle. A *preparation phase* spans several months before the event. Its outputs are a detailed game scenario, infrastructure deployed in the cyber range, scoring system, and game rules. A *dry run* phase involving testers in Blue teams helps to find flaws in the rules and the game scenario. Learners play the CDX game in an *execution phase*. An *evaluation phase* concludes the life-cycle. As a result, organizers use the outcomes in the next run. Data sources for the evaluation span from learners' feedback to automatically acquired data from the cyber range (e. g., computer logs, configuration changes, users' actions). More details about an exercise life cycle can be found in [244].

Knowledge of organizers of CDX is collective and continuous. Collective means that there are many organizers involved who share their experience to build and reuse the knowledge. Continuousness comes from the fact that methods of exploratory and confirmatory analyses used in the exploration loops of CDX usually produce approximate results leading to uncertain insight. Only repeating the analysis through multiple exercises can improve the insight by making it gradually more and more credible to be finally accepted as a piece of knowledge. Our goal is to adapt Sacha's VA framework for CDX so that we can build and share the knowledge systematically and efficiently via functionality provided by cyber ranges.

K.2 Visual Analytics in Cyber Exercises

Application of the VA framework on the organization of CDX requires clarifying the type of data, available models, visualizations, and also human-driven analytical processes depending on these computer-related elements. In what follows, we discuss the VA model from these individual perspectives and define a classification scheme that helps us to understand how the VA model fits requirements of CDX organizers.

K.2.1 Hypothesis-driven Analytical Goals

Hypotheses actively drive the unified VA model. Moreover, a vast amount of various teams and user roles involved in the organization can introduce a considerable amount of different objectives. These aspects could make the adoption of the unified VA framework and its systematic support in cyber ranges impossible.

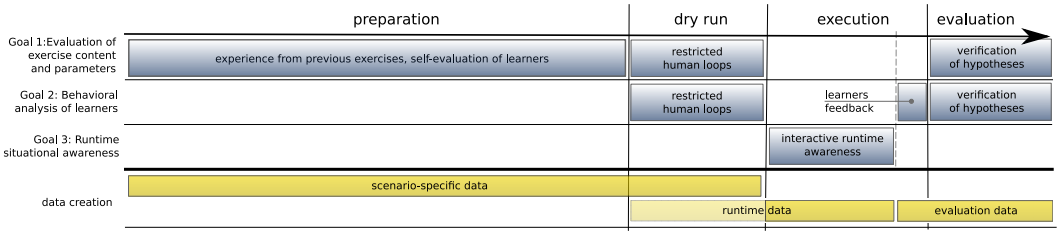

Figure K.2.1: Mapping of goals and data to phases of CDX life cycle.

To cope with these doubts, we divide common analytical goals to three distinct categories which enable us to (a) clarify the hypotheses, (b) classify them according to the goals, and (c) map verification loops of the hypotheses to individual phases of CDX life cycle. Figure K.2.1 overviews goals and their mapping to CDX life-cycle phases.

Goal 1 – Evaluation of exercise content and parameters

One of the most challenging tasks in the organization of cyber defense exercises is to make an exercise useful and to keep learners motivated to finish it. Therefore, hypotheses related to scenario difficulty, learners' confidence and satisfaction, learners' skills, and many other qualitative aspects, are often formulated.

During the *preparation phase*, organizers estimate and prepare key exercise parameters, such as a storyboard, a task schedule, penalty types and their values or types of attacks. Their improper values can make the exercise too complicated, too dull or unrealistic, which can quickly make learners frustrated. Therefore, organizers usually utilize a prior insight or knowledge gained from previous runs (results of their evaluation phase). Moreover, skills and experience of prospective learners are often ascertained employing self-evaluation questionnaires gathered in this phase. Results of this analytic loop are used to create well-balanced teams and to adapt exercise parameters to them.

Exercise parameters are tested and adjusted during the *dry run*. Note, that the verification loops are limited because participants involved in the dry run differ from learners and then also the results are approximate.

Hypotheses are verified during the *evaluation phase* when statistical models, knowledge-discovery models, exploratory visualizations, and other tactics of verification loops applied to exercise parameters and the data gathered during the exercise are brought into action. Gained insight and knowledge are used by organizers to prepare even better and more attractive exercises in the future.

Goal 2 – Behavioral analysis of learners

Study of the behavior of learners during an exercise can reveal relevant facts about their motivation, learning impact or level of knowledge. Gained information is useful for (a) learners as they can learn about themselves, their strengths, weaknesses, and mistakes; (b) exercise contractors, usually learners' employers, who can learn about the skills of their employees; (c) security experts and researchers who can reveal and compare atypical defense strategies, collaboration strategies, and other behavioral patterns. Therefore, organizers should be supported in these types of behavioral analyses so that they can verify behavior-related hypothesis and provide reasonable feedback to participating parties. Let us note that this goal is also partially related to the previous one because behavioral analysis of learners can also reveal problems caused by exercise parameters. For example, if organizers detect that several teams gave the exercise up in a particular phase of attacks plan, then they can infer insufficient difficulty of these attacks or inadequate readiness of learners.

The *dry run* is used to verify infrastructure, required data, and analytical loops. However, Blue teams involved in the testing are different from target learners. Neither the data nor possible analytical results are usually valid for gaining general knowledge, and they are erased before the execution phase.

At the end of the *execution phase*, it is convenient to provide feedback to learners so that they can analyze their behavior and learn from their mistakes immediately after the exercise. However, such feedback requires automatic data gathering and mediation to learners through intuitive and interactive analytical tools integrated into the cyber range. Adoption of the VA model by cyber ranges would help to achieve this valuable functionality.

The main effort related to the behavioral analysis is dedicated to the in-depth verification of corresponding hypotheses during the *evaluation phase*. The organizers present these results to learners during post-exercise workshops few weeks after the exercise.

Goal 3 – Runtime situational awareness

During an exercise, organizers monitor and analyze the situation on the "battlefield" and actively intervene if necessary. They have to analyze the situation from their perspective and interact with the system continuously. However, it is important to realize that runtime situational awareness provided to learners is intentionally very limited because in CDX the realism is of high importance. Therefore, giving an insight, which is not available in the real world, is undesirable. On the contrary, the goal of CDX is often to train learners in gaining the insight by themselves.

We can consider situational awareness as a process of making simple runtime hypotheses in the users' mind. The hypotheses are evaluated via interactive visual tools mediating access to the infrastructure and proving insight into its internal processes and developments. Interactions of learners produce data for the verification of organizers' hypotheses.

The situational awareness plays an essential role during the *execution phase*. It is not a passive process when a user is only notified of important events. On the contrary, actions and visualizations of situational awareness have to enable learners and organizers to interact with the system actively and then to affect its state. These interactions and changes in state are often monitored and used for following analytic tasks of the previous two analytical goals during the CDX evaluation phase.

K.2.2 Data

As hypotheses defined for cyber exercises are changing then also requirements on data are frequently changing. Moreover, CDX is often unstructured from the learner's perspective. For instance, a cyber range can generate network flows (data transmitted through networks), hosts and network characteristics (e. g., network throughput, memory size), system logs, questionnaires or scenario penalties. Variability and heterogeneity of data put high demands on the adaptability of the monitoring and storage infrastructure and make the design intriguing, as discussed in [6, 243, 181].

To clarify data required for visual analytics of CDX, we classify them according to the phases of exercise life cycle, as shown in Figure K.2.1. As classification criteria, we use data creation. However, it is worth to point out that data created in particular phase can be used to solve any analytical goal at any phase of the exercise life cycle.

Scenario-specific data

Configuration data defined by organizers usually in the preparation phase and possibly adjusted during the dry run. These data include, for example, a division of learners to teams, network topology, and network properties, the definition of required exercise services running on defended networks, types of penalties and their values or attack schedule. This category also includes answers to questions of various learners surveys.

Exercise runtime data

A system-generated data gathered and stored during the execution phase of an exercise. They represent quantitative operational data providing digital evidence of the behavior of users and applications during the exercise. Exercise runtime data is often based on the scenario-specific data and include, for example, particular penalty points assigned to teams, information about (un)availability of exercise services at given time or logs from hosts. Exercise runtime data is also collected in the dry run for testing purposes and then deleted.

Evaluation data

User-generated data provide qualitative information. This data is gathered either from learners at the end of exercise via post-exercise surveys, specialized feedback visualizations, or from organizers during the evaluation phase where additional data can be inserted to verify hypotheses. For example, structured informal notes about the behavior of learners noticed by organizers during the exercise can be added to the dataset.

K.2.3 Models

Models derived from data can be as simple as descriptive statistics or as complex as a data mining algorithms. Their usage is also reasonable in the context of cyber exercises. For example, complex networks [49] could be used to capture relationships between learners to simulate and analyze their behavioral patterns like collaboration or defense strategies. Knowledge discovery approaches to anomaly detection [149, 28, 10, 9] can reveal significant exercise parameters or learners with remarkable skills.

Nowadays, standard statistical models are used extensively for the evaluation of exercises [204, 173, 77, 108, 95]. On the contrary, the utilization of advanced models is exceptional and ad-hoc just because of missing conceptual solution to repeated analytical tasks in the CDX domain.

K.2.4 Visualizations

As for the visualizations, some classifications and perspectives allow us to cover different targets of existing models and available data. Our classification divides visualizations into three basic categories providing insight into data according to analytical goals.

Exercise infrastructure overview

Interactive visualizations that give us a complete overview of the structure and state of network infrastructure and help us to monitor running services. These visualizations are beneficial for runtime situational awareness (analytical Goal 3). However, the network topology overviews usually represent primary access points used by learners to interact with the infrastructure. Therefore, visualizations equipped with functions monitoring interactions of learners can help us to gather the data related to learners' behavior and then to verify hypotheses of the analytical Goal 2.

Visual insight into the exercise progression

Visualizations that aim at providing insight into the state and development of an exercise. Some insight can be gained from the discussed views on exercise infrastructure. For example, inaccessibility of services dues to a successful Red team's attack. However, it is not usually enough, and both learners and organizers need specialized views covering the exercise state. For example, Blue teams should be informed about the development of their score, while Red, Green, and White teams should have a detailed overview of planned and performed attacks and their successfulness so that they can distinguish between expected behavior and failures in the infrastructure, for instance, and then intervene properly.

This category of visualizations is useful primarily for the analytical Goal 3 because it provides situational awareness to both parties. At the same time, it can be helpful in verifying exercise parameters (Goal 1). For example, if the schedule of the exercise is not rich enough, or the Blue team is too busy or bored, the exercise parameters could be considered wrong and adjusted for future runs.

Feedback visualizations

The goal of visualizations providing interactive visual feedback is to gain insight into the exercise as well, but not from the perspective of current exercise progression. Instead, this insight is retrospective, aiming at learning from runtime mistakes, wrong decisions, or improperly estimated exercise parameters. Providing timely intuitive feedback to learners is crucial for improving the impact of the exercise. However, if the feedback is extended with the possibility to comment or rank events by the learners actively, then it can be even more useful. This kind of learners' reflection can help to reveal inappropriate exercise parameters (Goal 1) and to gather a data related to the behavior of individual learners (Goal 2).

K.3 Case Study

In this section, we illustrate the application of visual analytics process in CDX which we distilled from eight runs of Cyber Czech exercise series held in 2015–2017. Each run lasted two days and involved about 20 learners located in one physical place. We follow the exercise life cycle and put tasks and components of the visual analytics model into the context of individual phases so that their continuity is better recognizable. The iterative principle of visual analytics process results in multiple iterations over the refined and/or redefined hypotheses. Table K.3.1 summarizes results for primary hypotheses, as discussed in what follows.

Table K.3.1: Overview of the visual analytics process mapped on the initial hypotheses.

Human	**Hypothesis**	*The participants improve their skills*	*Every single participant is involved*	*Exercise infrastructure is stable and responsive enough to resemble realistic settings*
	Insight	Fairly confirmed. Individual learners would be affected by their skills and skills of teammates ⇒ hypotheses H1a and H1b. A novel ways of prerequisite testing are desired.	Fairly confirmed. Detailed per-user data required in the future. The level of involvement would stand as an indicator of cost-efficiency of the exercise ⇒ hypothesis H2a.	Uncertain results. Current views on monitoring data provide situational awareness but no statistics for evaluation of stability and responsiveness of the infrastructure.
	Actions	Organizers: Data definition, configuration of data sources (sub-systems) and visualizations, evaluation. Learners: Filling questionnaires, interaction with the cyber range and feedback visualizations.		
	Findings	Majority of the learners confirmed they learned new skills or re-shaped existing ones. Some learners did not learn anything new. Some others admitted the lack of necessary skills.	Majority of the learners declared they were involved. The data was, however, collected on a per-team basis. We were not able to objectively measure the level of involvement of every single participant.	A considerable amount of issues reported by learners. The Green team was aware of most of them. Several issues remained unnoticed.
Computer	**Data**	Data from scoring and auditing systems, pre- and post-exercise questionnaires.	Post-exercise questionnaires.	Data from the monitoring system, notes taken by organizers.
	Models	Descriptive statistics	Descriptive statistics	N/A
	Vis.	Feedback visualization	Feedback visualization	Nagios, network topology

K.3.1 Hypotheses

Hypotheses are either formulated during the preparation phase of a CDX or reused from previous exercises. We formulated several hypotheses, from which we selected the following we consider as the most important:

H1 – Participants improve their skills

First and foremost, the exercise should be useful for learners. It should deliver any educational value: either in technical, organizational or communication level. In particular, learners should develop or exercise skills required for incident handling and resolution, including reporting and communication with other parties outside their team, as well as working under stressful conditions. This hypothesis is related to the analytic Goal 1.

H2 – Every single participant is involved

Costs and effort invested in preparation and execution of the complex exercise should be utilized efficiently. Each learner should benefit from participating in the exercise. The content of the exercise should be rich enough to engage each participant. This hypothesis is related to Goals 1 and 2.

H3 – Exercise infrastructure is stable and responsive enough to resemble realistic settings

The CDX infrastructure is complicated. Multiple instances of separate environments of individual teams are deployed and transparently emulated on a restricted and complex infrastructure of the cyber range. However, any virtualization issue should not affect the user experience of real-life infrastructure regarding performance, response or failures. This hypothesis is related to the analytic Goals 1 and 3.

K.3.2 Preparation Phase

During the preparation phase, it is necessary to define data to be gathered for further analysis, configure data-related components of the cyber range, and prepare the graphical user environment. In particular, we perform the following preparation actions.

Preparation of surveys, formulation of questions

To verify our hypothesis, we use pre- and post-exercise questionnaires. This *evaluation data* are related to qualitative aspects of learners and exercise, e. g., participants' skills, their exercise experience, their opinion on difficulty or usability. We currently use the external Google Forms system to define and process questionnaires, which complicate the evaluation and integration of gained answers with internal data measured and stored in the cyber range.

Scoring subsystem settings

A scoring subsystem is used for penalization of Blue teams. Concrete penalties assigned to learners represent an *exercise runtime data* which are collected during the later phases of CDX life cycle. During this preparation phase, a *scenario-specific data* is used to define scoring rules. Attack plans, objectives, and their penalty values are set according to expected goals of the exercise and learners' skills.

Infrastructure monitoring settings

Green team members configure the infrastructure monitoring subsystem to keep track of the health of the virtualized networks and their underlying infrastructure. This step requires to specify a *scenario-specific data* like topology details, IP addresses of monitored hosts, network ports of watched services, or required timeouts. *Exercise runtime data* produced by the monitoring subsystem during the execution phase in the form of events is used for situational awareness of the Green team and for the automatized penalization of Blue teams for inaccessibility of network services (e. g., web, mail) that are under their management in the scenario. Currently, we use the Nagios monitoring system running in the cyber range as a standalone application. Its tighter "out of the box" integration into the cyber range would bring better connection to other internal data and then more effective situational awareness and analysis.

Configuration of auditing capabilities

While the infrastructure monitoring subsystem monitors infrastructure, the auditing subsystem monitors events that are related to the behavior of users and applications. This step includes the configuration of probes and internal auditing capabilities of the cyber range so that we can monitor required events like access to hosts, e-mail delivery, host reboots, or a history of commands run on a host by learners.

Configuration of runtime visualizations

Visualizations used in the cyber range are generic and highly configurable to cope with a wide variety of user goals. For example, the interactive network topology shows network-related *exercise runtime data* like current utilization of links or the state of nodes. However, this kind of situational awareness is undesirable for CDX, and the organizers have to adjust provided visualizations so that they satisfy specific requirements of the exercise.

K.3.3 Dry Run and Execution Phase

During the execution phase, interactions of various participants mingle. Moreover, the interactions reflect different levels and details of human loops of the visual analytics process. For instance, learners' actions produce data for exploration and verification loops of organizers. To discuss relevant activities meaningfully, we describe them from the viewpoints of the individual teams involved in the CDX.

Blue teams

In our exercise, the Blue teams produce data for verification loops of hypotheses *H1 – Participants improve their skills* and *H2 – Every single participant is involved*. Observations made by learners during their interactions with the network infrastructure lead to further interactions motivated to fulfill exercise tasks. Interactions are monitored and stored for verification loops of hypotheses of organizers. Moreover, learners also fill pre- and post-exercise questionnaires to evaluate their input knowledge and exercise experience respectively.

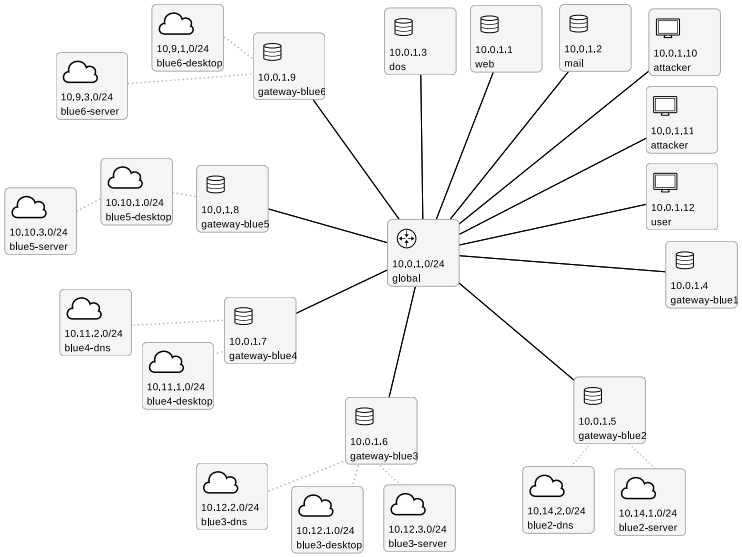

Figure K.3.1: Interactive network topology visualization.

During the exercise, learners use two runtime visualizations for situational awareness: network topology (see Figure K.3.1) and scoreboard. The former provides an overview of the network they administer and enables them to access individual hosts. The latter provides a score overview of all teams. The score includes both automatically collected data from the cyber range (e. g., availability of the web service or database) and inputs from other teams (answered questions of their users or journalists).

At the end of the exercise, learners get access to a specific visual-analytics tool for personalized feedback [242]. It displays the score development throughout the time enhanced with data points containing brief descriptions of reasons why score changed. These are coupled with analytical questions related to a retrospective evaluation of learners' actions by organizers, as shown in Figure K.3.2.

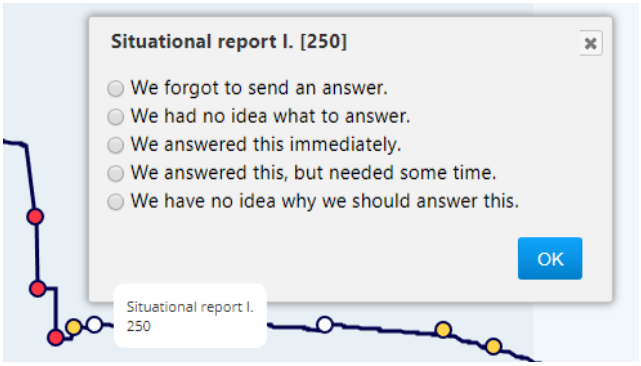

Figure K.3.2: An example of learners' evaluation of their actions

Green team

The Green team provides technical support during the exercise and produces data for *H3 – Exercise infrastructure is stable and responsive enough to resemble realistic settings.* The exercise execution is time-demanding since every issue needs to be solved as quickly as possible. Therefore, the more in-depth analysis of the issues and their solutions are mostly summarized in the evaluation phase.

The principal visual analytics tools of the Green team are network topology and Nagios dashboard. The network topology has the same capabilities as for the Blue teams, but it is rarely used. Most of the operations are done through the Nagios, which provides an overview of all Blue teams' networks. Also, it also displays service nodes that are not accessible by Blue teams, e. g., attackers' hosts.

Red team

Red team members perform the set of attacks according to the plan and enter penalty points based on the Blue teams' counter-actions. The gathered data about attacks and their successfulness is used to test H1 and H3 hypotheses.

The primary modus operandi is using a command line interface. The visualization tools are limited to a simple static schedule of the attacks made beforehand. For gaining better game situational awareness, the Red team needs to cooperate closely with the green one. For instance, to assess the success of their attacks, they need to know whether the attack was unsuccessful due to an adequate counter-action of a Blue team or because of an outage of the cyber range.

White team

While the Red team focuses on *hard-skills*, White team members enter penalties based on *soft-skills* findings. They provide data for testing H1 and H2 hypotheses. They need to know the state of the game since some of their injects are time-related to attacks of the Red team. Therefore, they need to cooperate with both green and Red teams. Unfortunately, we have no tool for this type of orchestration and situational awareness available nowadays. The teams have to synchronize their activities via external communication channels.

Besides other activities, the White team also play a role of ordinary users. To do that, members of the team use the topology visualization to access hosts, simulate frequent utilization of the network, and interact with Blue teams.

K.3.4 Evaluation phase

Main outcomes of the evaluation phase are hypotheses for the second iteration of the visual analytics process. These are based on the findings and insight gained from the verification loops of the original set of hypotheses:

H1 – The participants improve their skills

We analyzed learners' pre- and post-exercise surveys and exercise scores using standard statistical models and exploration loops. We found out that the majority of learners confirmed they learned new skills or re-shaped existing ones. However, some learners reported that they did not learn anything new and some others admitted they lack some necessary prerequisite skills. Both extremes may indicate flaws in the selection of individual learners, their grouping to a team, or structure and content of exercise tasks.

H2 – Every single participant is involved

The vast majority of collected exercise runtime data captures actions of teams, not individual learners. The only data sources we analyzed to verify this hypothesis were post-exercise surveys and reflections provided by individual learners using an application providing automatically generated feedback visualizations. Although the results indicate that almost all participants felt involved during the exercise, the level of their involvement is unknown and may vary widely due to the absence of objective and rich data tracking all available modes of interactions.

H3 – Exercise infrastructure is stable and responsive enough to resemble realistic settings

Exercise runtime data serve for determining the exercise score and monitoring the exercise infrastructure. The current view of the data does not provide any statistics for evaluation of stability and responsiveness of exercise infrastructure. The only data sources are learners' post-exercise surveys and notes of issues taken by members of teams of organizers, in particular by Green team. A considerable amount of learners reported various issues. The organizers were fully aware of some of them, but there were several issues unnoticed. Again, the objective data source would help to clarify this bias.

K.3.5 Derived Hypotheses

Consequent hypotheses derived during the iterative VA process usually emerge. Due to the space restrictions of the paper, we end up with the outline of the hypotheses for the second iteration to illustrate the process continuation.

H1a – The difficulty of the exercise was adequate for learners

One of our observations related to H1 is that cohorts of dry-run and execution participants bias the perception of the difficulty level. The input knowledge of actual learners (not testers) needs to be considered. However, the pre-exercise survey relies only on self-assessment of learners' skills that could introduce an unwanted bias. We advocate complementing the self-assessment survey by a quiz or a practical task that would test the required skills objectively. As a result, prospective participants would have skills adequate to the exercise difficulty. The hypothesis relates to the analytic Goals 1 and 2.

H1b – Learners form well-balanced teams

While the CDX is based firmly on teamwork, grouping people of different skills into well-performing team covering as much as prerequisite skills is crucial. Weaknesses of one shall be balanced by strengths of another member of the team and vice versa. Since the teams are formed before the exercise, the pre-exercise survey questions should be refined to acquire more accurate data for optimal team balancing. The hypothesis relates to the analytic Goal 2.

H2a – Participants' involvement stands as an indicator for cost-efficiency of the exercise

CDX is a costly event. While person-months spent and costs of computational resources can be calculated relatively easily, answering the question whether the costs are adequate is tough. Participants' involvement could be a good indicator. The organizers should be able to determine the involvement ratio for each participant as well as the overall involvement of the whole group. The methodology for gaining the involvement ratio should combine outcomes from post-exercise questionnaires with analysis of participants' behavior and actions (e. g., from the automatically collected logs and commands they entered). As a side effect, the organizers should be able to provide learners with personalized feedback on their strengths and weaknesses. The hypothesis relates to the analytic Goals 1 and 2.

H2b: The set of exercise tasks covers relevant security issues

Thousands of threats exist, but only a subset of them is relevant these days. Attacks selected for the exercise should exploit recent and relevant threats rather than out-of-date and insignificant ones. While the obsolete threats can be suitable for the educational purpose, the organizers should carefully consider and select those, that are relevant nowadays (i. e., participants can experience them in their work). Strongly outdated threats (e. g., those that focus on no more used version of an operating system or a web server) are inappropriate. Common Vulnerability Scoring System (CVSS) by FIRST[9] can be used to assess the relevance of the threats. The hypothesis relates to the analytic Goal 1.

K.3.6 Output Knowledge

Verification of hypotheses H1–H3 brought a valuable insight regarding the usability of our cyber range, attractiveness for learners and the level of impact on them. However, the total number of teams that have been involved in the exercise series and provided data for the verification is relatively small. For this reason, we perceive conclusions formulated in this section as insight only. To be able to declare them as a justified knowledge, we need to verify them on more runs.

K.4 Lessons Learned

Organization of complex CDX is usually ad-hoc hence inefficient

Modern cyber ranges support the organization of complex CDX. However, the organization discussed in Section K.1.2 requires a vast amount of manual work and interventions in the

[9] https://www.first.org/cvss/

infrastructure. Data useful for the optimization of CDX organization and improvement of exercise experience is often not gathered at all, or the data processing is not systematical. The data is usually exported manually from internal data sources and then processed in external tools ex-post. Our goal is to organize CDX efficiently, to use data as soon as possible (often at runtime), and to evaluate the impact on learners and the overall quality regularly. The classification of analytical tasks and their visual analytics elements discussed in this paper in the context of CDX life cycle would help us to solve this goal by identifying and clarifying processes that can be systematically supported by the cyber range.

Hypothesis-centered approach to CDX is suitable

Hypotheses actively drive the visual analytics model used as a unified framework for our approach. Although we were not using this hypothesis-centered way of thinking during the realization of previous exercises intentionally, we have come to realize that in fact, we were thinking in this way intuitively in many cases. Moreover, we found this kind of mindset handy for the definition of required data and the design of supporting interactive visual tools during the preparation of new exercises.

Positive impact on learners and organizers

Integration of even a few preliminary features of the visual analytics process into our cyber range brought positive outcomes from both learners and organizers, as shown in [242]. The application of the VA process to the organization of CDX also encouraged us to formalize attack plans, objectives, and other scenario-related events. Consequently, they are used for systematic analysis and runtime coordination of Green, Blue, White, and Red teams.

Structure of CDX-related knowledge was clarified

Nowadays, organizers have defined several processes prescribing how cyber defense exercises and their validation results should be documented and shared among team members so that exercises can be continuously adjusted and improved. However, the documentation is informal. It was not clear so far what the CDX-related knowledge exactly means and how to structure the pieces of information. Classification of CDX processes and elements discussed in this paper brings clear terminology and semantics which are suitable for formal knowledge modeling, e. g., using formal ontologies.

K.5 Conclusions and Future Work

Cyber defense exercises are complex education events requiring a significant amount of efforts of interdisciplinary teams. Application of the visual analytics process proves beneficial to

CDX organization and evaluation. As we demonstrated on our case study, the iterative approach of human loop helps us in the identification of issues and leads us towards concrete suggestions for improvements in the organization of CDX. Simultaneously, application of the visual analytics process clarified the structure of the CDX-related knowledge enabling its better management.

However, we need to explore and revise the VA components further. While having raw data from the exercises, we have an unclear notion about the valuable data models applicable in this domain. The useful visualizations are alike. The lack of VA tools integrated into cyber ranges is a severe weakness of nowadays. These tools could provide automated statistical analysis as well as more in-depth insight into the learner's behavior during the game. They could also help in improving the process of CDX organization.

In this paper, we focus mainly on the organizers' viewpoint. Learners could also apply the VA process even though they have entirely different experience than organizers. They are focused on particular tasks related to the exercise content rather than the overall process. For this reason, we leave this topic for our future work.

Acknowledgments

This research was supported by the Security Research Programme of the Czech Republic 2015–2020 (BV III/1 – VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014 – Simulation, detection, and mitigation of cyber threats endangering critical infrastructure.

Access to the CERIT-SC computing and storage facilities provided under the programme Center CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, reg. no. CZ. 1.05/3.2.00/08.0144, is greatly appreciated.

Bibliography

[1] Avatao. `https://avatao.com`. Accessed: 2017-05-22.

[2] Cyber ranges. `https://www.nist.gov/sites/default/files/documents/2017/05/23/cyber_ranges_2017.pdf`. Accessed: 2017-05-22.

[3] Cyber virtual ad hoc network (CyberVAN). `http://www.appcomsci.com/research/tools/cybervan`. Accessed: 2017-05-22.

[4] EDURange. `http://www.edurange.org`. Accessed: 2017-05-22.

[5] System & software process engineering metamodel (spem) 2.0. Technical report, Object Management Group, Inc, 2008.

[6] R. G. Abbott, J. McClain, B. Anderson, K. Nauer, A. Silva, and C. Forsythe. Log Analysis of Cyber Security Training Exercises. *Procedia Manufacturing*, 3:5088–5094, 2015. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

[7] C. C. Abt. *Serious Games*. University Press of America, 1987.

[8] C. N. Adams and D. H. Snider. Effective Data Visualization in Cybersecurity. In *SoutheastCon 2018*, pages 1–8. IEEE, 2018.

[9] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19 – 31, 2016.

[10] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, May 2015.

[11] H. Alliance. CyberRX 2.0 Level I Playbook Participant and Facilitator Guide. Technical report, HITRUST Alliance, LLC, 2015.

[12] C. Alonso-Fernandez, A. Calvo, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon. Systematizing game learning analytics for serious games. In *2017 IEEE global engineering education conference (EDUCON)*, pages 1111–1118, New York, 2017. IEEE, IEEE.

[13] R. Anderson, C. Barton, R. Boehme, R. Clayton, C. Ganan, M. Levi, T. Moore, and M. Vasek. Measuring the Cost of Cybercrime. In *Proceedings of the 18th Annual Workshop on the Economics of Information Security*, 2019.

[14] D. Arendt, D. Best, R. Burtner, and C. Lyn Paul. CyberPetri at CDX 2016: Real-time network situation awareness. In *2016 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–4, New York, 2016. IEEE Press.

[15] S. Arnab, T. Lim, M. B. Carvalho, F. Bellotti, S. De Freitas, S. Louchart, N. Suttie, R. Berta, and A. De Gloria. Mapping Learning and Game Mechanics for Serious Games Analysis. *British Journal of Educational Technology*, 46(2):391–411, 2015.

[16] A. Arnes, P. Haas, G. Vigna, and R. A. Kemmerer. Using a virtual security testbed for digital forensic reconstruction. *Journal in Computer Virology*, 2(4):275–289, 2007.

[17] A. E. Attipoe, J. Yan, C. Turner, and D. Richards. Visualization Tools for Network Security. *Electronic Imaging*, 2016(1):1–8, 2016.

[18] A. Bangor, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123, May 2009.

[19] Y. Bar-Yam. When systems engineering fails-toward complex systems engineering. In *2003 IEEE International Conference on Systems, Man and Cybernetics (SMC'03)*, volume 2, pages 2021–2028. IEEE, 2003.

[20] G. Bassett. System and Method for Cyber Security Analysis and Human Behavior Prediction, Mar. 22 2016. US Patent 9,292,695.

[21] G. Baumgarten, M. Rosinger, A. Todino, and R. de Juan Marín. SPEM 2.0 as process baseline Meta-Model for the development and optimization of complex embedded systems. In *2015 IEEE International Symposium on Systems Engineering (ISSE)*, pages 155–162. IEEE, 2015.

[22] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta. Assessment in and of Serious Games: An Overview. *Adv. in Hum.-Comp. Int.*, 2013:1:1–1:1, Jan. 2013.

[23] T. Benzel. The Science of Cyber Security Experimentation: The DETER Project. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 137–148. ACM, 2011.

[24] M. Beran, F. Hrdina, D. Kouřil, R. Ošlejšek, and K. Zákopčanová. Exploratory analysis of file system metadata for rapid investigation of security incidents. In *2020 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 11–20. IEEE Computer Society, 2020.

[25] E. Bertini and D. Lalanne. Surveying the Complementary Role of Automatic Data Analysis and Visualization in Knowledge Discovery. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, VAKD '09, pages 12–20, New York, NY, USA, 2009. ACM.

[26] D. M. Best, A. Endert, and D. Kidwell. 7 Key Challenges for Visualization in Cyber Network Defense. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 33–40. ACM, 2014.

[27] R. Beuran, D. Tang, C. Pham, K.-i. Chinen, Y. Tan, and Y. Shinoda. Integrated framework for hands-on cybersecurity training: CyTrONE. *Computers & Security*, 78:43–59, 2018.

[28] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys Tutorials*, 16(1):303–336, First 2014.

[29] A. T. Bimba, N. Idris, A. Al-Hunaiyyan, R. B. Mahmud, A. Abdelaziz, S. Khan, and V. Chang. Towards knowledge modeling and manipulation technologies: A survey. *International Journal of Information Management*, 36(6):857–871, 2016.

[30] R. Bodily and K. Verbert. Review of Research on Student-Facing Learning Analytics Dashboards and Educational Recommender Systems. *IEEE Trans. on Learning Tech.*, 10(4):405–418, Oct 2017.

[31] R. Bodily and K. Verbert. Trends and issues in student-facing learning analytics reporting systems research. In *Proceedings of the seventh international learning analytics & knowledge conference*, pages 309–318. ACM, 2017.

[32] B. Boehm and R. Turner. *Balancing agility and discipline: A guide for the perplexed.* Addison-Wesley Professional, 2003.

[33] K. Boopathi, S. Sreejith, and A. Bithin. Learning Cyber Security Through Gamification. *Indian Journal of Science and Technology*, 8(7):642–649, 2015.

[34] A. Boschetti, L. Salgarelli, C. Muelder, and K.-L. Ma. TVi: a visual querying system for network monitoring and anomaly detection. In *Proceedings of the 8th international symposium on visualization for cyber security*, pages 1–10, 2011.

[35] V. Botta-Genoulaz, P.-A. Millet, and B. Grabot. A survey on the recent research literature on ERP systems. *Computers in industry*, 56(6):510–522, 2005.

[36] A. Brilingaitė, L. Bukauskas, and A. Juozapavišius. A framework for competence development and assessment in hybrid cybersecurity exercises. *Computers & Security*, page 101607, 2019.

[37] F. Buchholz and E. Spafford. On the role of file system metadata in digital forensics. *Digital Investigation*, 1(4):298 – 309, 2004.

[38] F. P. Buchholz and C. Falk. Design and Implementation of Zeitline: a Forensic Timeline Editor. In *Proceedings of the fifth annual DRFWS Conference*, 2005.

[39] L. Buttyán, M. Félegyházi, and G. Pék. Mentoring talent in IT security–A case study. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, 2016.

[40] N. Cao, L. Lu, Y.-R. Lin, F. Wang, and Z. Wen. Socialhelix: visual analysis of sentiment divergence in social media. *Journal of Visualization*, 18(2):221–235, 2015.

[41] N. Cao, C. Shi, S. Lin, J. Lu, Y. Lin, and C. Lin. Targetvue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):280–289, Jan 2016.

[42] B. Cappers. *Interactive visualization of event logs for cybersecurity.* PhD thesis, Department of Mathematics and Computer Science, Dec. 2018. Proefschrift.

[43] B. Carrier. *File System Forensic Analysis.* Addison-Wesley Professional, 2005.

[44] E. Casey. *Handbook of Digital Forensics and Investigation.* Academic Press, Inc., 2009.

[45] L. Caviglione, S. Wendzel, and W. Mazurczyk. The Future of Digital Forensics: Challenges and the Road Ahead. *IEEE Security & Privacy*, 15(6):12–17, 2017.

[46] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[47] P. Chapman, J. Burket, and D. Brumley. PicoCTF: A Game-Based Computer Security Competition for High School Students. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, San Diego, CA, 2014. USENIX Association.

[48] S. Charleer, A. V. Moere, J. Klerkx, K. Verbert, and T. De Laet. Learning Analytics Dashboards to Support Adviser-Student Dialogue. *IEEE Trans. on Learning Tech.*, 11(3):389–399, July 2018.

[49] G. Chen, X. Wang, and X. Li. *Fundamentals of complex networks: models, structures and dynamics.* John Wiley & Sons, 2014.

[50] L. Chen. Construction of the New Generation Network Security Testbed-Testbed@ TWISC: Integration and Implementation on Software Aspect, 2008. Institute of Computer & Communication, National Cheng Kung University, Tainan, Taiwan.

[51] N. Childers, B. Boe, L. Cavallaro, L. Cavedon, M. Cova, M. Egele, and G. Vigna. Organizing large scale hacking competitions. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 132–152. Springer, 2010.

[52] G. K. Chung. Guidelines for the design and implementation of game telemetry for serious games analytics. In *Serious games analytics*, pages 59–79. Springer, Berlin, Heidelberg, 2015.

[53] K. Chung. Live Lesson: Lowering the Barriers to Capture The Flag Administration and Participation. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*, Vancouver, BC, 2017. USENIX Association.

[54] Cisco Systems. Cisco 2014 annual security report. `http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf`, 2014. Accessed: 2018-08-08.

[55] L. Corrin and P. de Barba. How do students interpret feedback delivered via dashboards? In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, pages 430–431, New York, NY, USA, 2015. ACM.

[56] CTF365. Capture the flag 365. `https://ctf365.com`. Accessed: 2017-05-22.

[57] CTFtime team. CTFtime.org / All about CTF (Capture The Flag). `https://ctftime.org/`. (Accessed on 2018-06-13).

[58] D. Dasgupta, D. M. Ferebee, and Z. Michalewicz. Applying Puzzle-Based Learning to Cyber-Security Education. In *Proceedings of the 2013 on InfoSecCD '13: Information Security Curriculum Development Conference*, InfoSecCD '13, pages 20:20–20:26, New York, NY, USA, 2013. ACM.

[59] A. Davis, T. Leek, M. Zhivich, K. Gwinnup, and W. Leonard. The fun and future of ctf. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, San Diego, CA, 2014. USENIX Association.

[60] J. Davis and S. Magrath. A Survey of Cyber Ranges and Testbeds. Technical Report DSTO-GD-0771, Defence Science and Technology Organisation: Cyber and Electronic Warfare Division, 2013.

[61] S. de Freitas, D. Gibson, V. Alvarez, L. Irving, K. Star, S. Charleer, and K. Verbert. How to use gamified dashboards and learning analytics for providing immediate student feedback and performance tracking in higher education. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 429–434, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[62] F. A. Deeb and T. Hickey. Classroom orchestration with problem solving markov models. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, Oct 2018.

[63] S. Diaw, M. L. Cisse, and A. Bah. Using the SPEM 2.0 kind-based extension mechanism to define the SPEM4MDE metamodel. In *Proceedings of the International Conference on Computing for Engineering and Sciences*, pages 63–69. ACM, 2017.

[64] A. Diehl, L. Pelorosso, C. Delrieux, C. Saulo, J. Ruiz, M. E. Gröller, and S. Bruckner. Visual analysis of spatio-temporal data: Applications in weather forecasting. *Computer Graphics Forum*, 34(3):381–390, May 2015.

[65] A. Diehl, L. Pelorosso, K. Matkovic, J. Ruiz, M. E. Gröller, and S. Bruckner. Albero: A visual analytics approach for probabilistic weather forecasting. *Computer Graphics Forum*, 36(7):135–144, Oct. 2017.

[66] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe. A decade of agile methodologies: Towards explaining agile software development, 2012.

[67] A. Doupé, M. Egele, B. Caillat, G. Stringhini, G. Yakin, A. Zand, L. Cavedon, and G. Vigna. Hit 'em where it hurts: A live security exercise on cyber situational awareness. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 51–61, New York, NY, USA, 2011. ACM.

[68] K. Dočkalová Burská, V. Rusňák, and R. Ošlejšek. Data-driven insight into the puzzle-based cybersecurity training. *Computers & Graphics*, to appear, 2021.

[69] K. Dočkalová Burská, V. Rusňák, and R. Ošlejšek. Enhancing situational awareness for tutors of cybersecurity capture the flag games. In *25th International Conference Information Visualisation (IV).*, pages 236–243. IEEE Computer Society, 2021.

[70] D. Duchamp and G. De Angelis. A hypervisor based security testbed. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007*, DETER, Berkeley, CA, USA, 2007. USENIX Association.

[71] E. G. Díez, D. F. Pereira, M. A. L. Merino, H. R. Suárez, and D. B. Juan. Cyber exercises taxonomy. Technical report, INCIBE, 2015.

[72] A. D'Amico, L. Buchanan, D. Kirkpatrick, and P. Walczak. Cyber Operator Perspectives on Security Visualization. In *Advances in Human Factors in Cybersecurity*, pages 69–81. Springer, 2016.

[73] C. Eagle. Computer Security Competitions: Expanding Educational Outcomes. *IEEE Security & Privacy*, 11(4):69–71, 2013.

[74] V. Echeverria, R. Martinez-Maldonado, R. Granda, K. Chiluiza, C. Conati, and S. B. Shum. Driving data storytelling from learning design. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 131–140. ACM, 2018.

[75] Z. Eichler, R. Ošlejšek, and D. Toth. Kypo: A tool for collaborative study of cyberattacks in safe cloud environment. In *HCI International 2015: Human Aspects of Information Security, Privacy, and Trust*, pages 190–199, Los Angeles, 2015. Springer International Publishing.

[76] Emulab. A list of Emulab Testbeds. `http://wiki.emulab.net/Emulab/wiki/OtherEmulabs`. Accessed: 2017-05-22.

[77] A. Endert, C. Han, D. Maiti, L. House, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.

[78] M. R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, 37(1):32–64, 1995.

[79] M. R. Endsley. *Designing for Situation Awareness: An Approach to User-centered Design*. CRC press, 2016.

[80] ENISA. Cyber Europe 2016 – After action report : findings from a cyber crisis exercise in Europe. Technical report, European Union Agency for Network and Information Security, 2016.

[81] T. Faber and J. Wroclawski. A federated experiment environment for Emulab-based testbeds. In *TRIDENTCOM*, pages 1–10, 2009.

[82] A. Falah, L. Pan, and M. Abdelrazek. Visual representation of penetration testing actions and skills in a technical tree model. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '17, pages 8:1–8:10, New York, NY, USA, 2017. ACM.

[83] B. Ferguson, A. Tall, and D. Olsen. National cyber range overview. In *2014 IEEE Military Communications Conference*, pages 123–128, Oct 2014.

[84] E. E. Fırat and R. S. Laramee. Towards a Survey of Interactive Visualization for Education. *EG UK Computer Graphics & Visual Computing*, 2018.

[85] E. Fouh, M. Akbar, and C. A. Shaffer. The Role of Visualization in Computer Science Education. *Computers in the Schools*, 29(1-2):95–117, 2012.

[86] M. Fowler. *Analysis Patterns: Reusable Object Models*. Addison-Wesley Professional, 1997.

[87] X. Fu, A. Shimada, H. Ogata, Y. Taniguchi, and D. Suehiro. Real-time learning analytics for c programming language courses. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 280–288, New York, NY, USA, 2017. ACM.

[88] S. Galván-Cruz, M. Mora, and R. O'Connor. A means-ends design of scrum+: an agile-disciplined balanced scrum enhanced with the iso/iec 29110 standard. In *International Conference on Software Process Improvement*, pages 13–23. Springer, 2017.

[89] Gartner, Inc. Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019. `https://muni.cz/go/c7a9e9`, August 2018.

[90] S. A. K. Ghayyur, S. Ahmed, M. Ali, A. Razzaq, N. Ahmed, and A. Naseem. A systematic literature review of success factors and barriers of agile software development. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(3), 2018.

[91] J. C. Gibbs and J. D. Taylor. Comparing student self-assessment to individualized instructor feedback. *Active Learning in Higher Education*, 17(2):111–123, 2016.

[92] M. Gondree, Z. N. Peterson, and T. Denning. Security Through Play. *IEEE Security & Privacy*, 11(3):64–67, 2013.

[93] S. Govaerts, K. Verbert, E. Duval, and A. Pardo. The Student Activity Meter for Awareness and Self-reflection. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 869–884, New York, NY, USA, 2012. ACM.

[94] S. Govaerts, K. Verbert, J. Klerkx, and E. Duval. Visualizing Activities for Self-reflection and Awareness. In *International Conference on Web-based Learning*, pages 91–100. Springer, 2010.

[95] M. Granåsen and D. Andersson. Measuring team effectiveness in cyber-defense exercises: a cross-disciplinary case study. *Cognition, Technology & Work*, 18(1):121–143, Feb 2016.

[96] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE transactions on visualization and computer graphics*, 19(12):2277–2286, 2013.

[97] C. C. Gray, P. D. Ritsos, and J. C. Roberts. Contextual network navigation to provide situational awareness for network administrators. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8. IEEE, 2015.

[98] S. Gregor and A. R. Hevner. Positioning and presenting design science research for maximum impact. *MIS quarterly*, pages 337–355, 2013.

[99] S. Grissom, M. F. McNally, and T. Naps. Algorithm visualization in cs education: Comparing levels of student engagement. In *Proceedings of the 2003 ACM Symposium on Software Visualization*, SoftVis '03, pages 87–94, New York, NY, USA, 2003. ACM.

[100] D. Hall. *Ansible configuration management*. Packt Publishing Ltd, 2013.

[101] B. Hallaq, A. Nicholson, R. Smith, L. Maglaras, H. Janicke, and K. Jones. Cyran: a hybrid cyber range for testing security on ics/scada systems. In *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications*, pages 622–637. IGI Global, 2018.

[102] B. Hanington and B. Martin. *Universal Methods of Design: 100 ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions*. Rockport Publishers, 2012.

[103] C. Hargreaves and J. Patterson. An automated timeline reconstruction approach for digital forensic investigations. *Digital Investigation*, 9:S69 – S79, 2012.

[104] K. J. Harms, N. Rowlett, and C. Kelleher. Enabling Independent Learning of Programming Concepts Through Programming Completion Puzzles. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 271–279. IEEE, 2015.

[105] J. A. Hattie, G. T. Brown, L. Ward, S. E. Irving, and P. J. Keegan. Formative Evaluation of an Educational Assessment Technology Innovation: Developers' Insights into Assessment Tools for Teaching and Learning (asTTle). *Journal of Multi-Disciplinary Evaluation*, 5(3):1–54, 2006.

[106] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective visualization of file system access-control. In *International Workshop on Visualization for Computer Security*, pages 18–25. Springer, 2008.

[107] M. Hendrix, A. Al-Sherbaz, and B. Victoria. Game-based Cyber Security Training: Are Serious Games Suitable for Cyber Security Training? *International Journal of Serious Games*, 3(1):53–61, 2016.

[108] D. S. Henshel, G. M. Deckard, B. Lufkin, N. Buchler, B. Hoffman, P. Rajivan, and S. Collman. Predicting proficiency in cyber defense team exercises. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 776–781, Nov 2016.

[109] R. Hoda, N. Salleh, and J. Grundy. The rise and evolution of agile software development. *IEEE Software*, 35(5):58–63, 2018.

[110] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *Communications Surveys Tutorials, IEEE*, PP(99):1–1, 2014.

[111] K. Holstein, G. Hong, M. Tegene, B. M. McLaren, and V. Aleven. The Classroom as a Dashboard: Co-Designing Wearable Cognitive Augmentation for K-12 Teachers. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 79—88, New York, NY, USA, 2018. Association for Computing Machinery.

[112] K. Holstein, B. M. McLaren, and V. Aleven. Intelligent Tutors as Teachers' Aides: Exploring Teacher Needs for Real-Time Analytics in Blended Classrooms. In *Proc. of the Seventh Int. Learning Analytics & Knowledge Conference*, LAK '17, pages 257–––266, New York, NY, USA, 2017. Association for Computing Machinery.

[113] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang. TrajGraph: A Graph-based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):160–169, 2016.

[114] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. Elvis: Extensible log visualization. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pages 9–16, 2013.

[115] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. Corgi: Combination, organization and reconstruction through graphical interactions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 57–64, 2014.

[116] T. Imani, M. Nakano, and V. Anantatmula. Does a hybrid approach of agile and plan-driven methods work better for it system development projects? *development*, 1(2):3, 2017.

[117] Societal security – guidelines for exercises. Standard, International Organization for Standardization, Geneva, CH, Sept. 2013.

[118] K. L. Jacobs. Investigation of Interactive Online Visual Tools for the Learning of Mathematics. *International Journal of Mathematical Education in Science and Technology*, 36(7):761–768, 2005.

[119] T. Jirsík, M. Husák, P. Čeleda, and Z. Eichler. Cloud-based security research testbed: A ddos use case. In H. Lutfiyya and P. Cholda, editors, *Proceedings of the Network Operations and Management Symposium (NOMS 2014)*, Krakow, Poland, 2014. IEEE Xplore Digital Library.

[120] I. Jivet, M. Scheffel, M. Specht, and H. Drachsler. License to evaluate: Preparing learning analytics dashboards for educational practice. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 31–40, New York, NY, USA, 2018. ACM.

[121] S. Kälber, A. Dewald, and F. C. Freiling. Forensic Application-Fingerprinting Based on File System Metadata. In *Proceedings of the IEEE 2013 Seventh International Conference on IT Security Incident Management and IT Forensics*, pages 98–112, 2013.

[122] J. Kävrestad. *Fundamentals of Digital Forensics: Theory, Methods, and Real-Life Applications*. Springer International Publishing, 2018.

[123] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. Mastering the information age solving problems with visual analytics. In *Eurographics*, volume 2, page 5, 2010.

[124] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In *Visual data mining*, pages 76–90. Springer, 2008.

[125] J. Kick. Cyber Exercise Playbook. Technical report, MITRE Corp., Bedford, MA, 2014.

[126] L. C. Koh, A. Slingsby, J. Dykes, and T. S. Kam. Developing and Applying a User-centered Model for the Design and Implementation of Information Visualization Tools. In *15th Int. Conf. on Information Vis.*, pages 90–95. IEEE, 2011.

[127] M. Kont, M. Pihelgas, K. Maennel, B. Blumbergs, and T. Lepik. Frankenstack: Toward real-time red team feedback. In *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*, pages 400–405. IEEE, 2017.

[128] D. Kouřil, T. Rebok, T. Jirsík, J. Čegan, M. Drašar, M. Vizvěry, and J. Vykopal. Cloud-based testbed for simulation of cyber attacks. In H. Lutfiyya and P. Cholda, editors, *Proceedings of the Network Operations and Management Symposium (NOMS 2014)*, Krakow, Poland, 2014. IEEE Xplore Digital Library.

[129] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum*, n/a(n/a):n/a–n/a, 2016.

[130] S. Kriglstein, M. Pohl, S. Rinderle-Ma, and M. Stallinger. Visual Analytics in Process Mining: Classification of Process Mining Techniques. In *EuroVis Workshop on Visual Analytics (EuroVA). The Eurographics Association*, 2016.

[131] K. Krishna, W. Sun, P. Rana, T. Li, and R. Sekar. V-NetLab: a cost-effective platform to support course projects in computer security. In *Proceedings of 9th Colloquium for Information Systems Security Education*, 2005.

[132] M. Krone, B. Kozlikova, N. Lindow, M. Baaden, D. Baum, J. Parulek, H.-C. Hege, and I. Viola. Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum*, 2016.

[133] P. Kruchten. *The rational unified process: an introduction.* Addison-Wesley Professional, 2004.

[134] S. Kumar, R. Zafarani, and H. Liu. Understanding user migration patterns in social media. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[135] KYPO. `https://kypo.cz/`. Accessed: 2017-05-16.

[136] C. Larman and V. R. Basili. Iterative and incremental developments. a brief history. *Computer*, 36(6):47–56, 2003.

[137] K. Lawonn, N. N. Smit, K. Bühler, and B. Preim. A Survey on Multimodal Medical Data Visualization. 37(1):413–438, 2018.

[138] D. Leony, A. Pardo, L. de la Fuente Valentín, D. S. de Castro, and C. D. Kloos. Glass: A learning analytics visualization tool. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 162–163, New York, NY, USA, 2012. ACM.

[139] T. R. Leschke and C. Nicholas. Change-link 2.0: a digital forensic tool for visualizing changes to shadow volume data. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pages 17–24, 2013.

[140] Linux Foundation. Open vSwitch. `http://openvswitch.org/`. Accessed: 2017-05-22.

[141] C. S. Loh. Information trails: In-process assessment of game-based learning. In *Assessment in game-based learning*, pages 123–144. Springer, Berlin, Heidelberg, 2012.

[142] C. S. Loh, Y. Sheng, and D. Ifenthaler. *Serious Games Analytics: Methodologies for Performance Measurement, Assessment, and Improvement*. Advances in Game-Based Learning. Springer International Publishing, 2015.

[143] J. Ma, J. Tao, J. Mayo, C.-K. Shene, M. Keranen, and C. Wang. *AESvisual: A Visualization Tool for the AES Cipher*, page 230–235. Association for Computing Machinery, New York, NY, USA, 2016.

[144] L. P. Macfadyen and S. Dawson. Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & education*, 54(2):588–599, 2010.

[145] L. P. Macfadyen and S. Dawson. Mining lms data to develop an "early warning system" for educators: A proof of concept. *Computers & education*, 54(2):588–599, 2010.

[146] K. Maennel. Improving and Measuring Learning Effectiveness at Cyber Defence Exercises. Master's thesis, University of Tartu, Institute of Computer Science, 5 2017.

[147] K. Maennel, R. Ottis, and O. Maennel. Improving and Measuring Learning Effectiveness at Cyber Defense Exercises. In *Nordic Conference on Secure IT Systems*, pages 123–138. Springer, 2017.

[148] Mandiant Corp. Exposing one of China's cyber espionage units – Mandiant APT1 report. `http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf`, 2013. Accessed: 2018-08-08.

[149] G. Mariscal, Ó. Marbán, and C. Fernández. A survey of data mining and knowledge discovery process models and methodologies. *Knowledge Eng. Review*, 25(2):137–166, 2010.

[150] S. Mäses et al. Obtaining Better Metrics for Complex Serious Games Within Virtualised Simulation Environments. In *European Conference on Games Based Learning*, pages 428–434, 2017.

[151] W. Matcha, D. Gašević, N. A. Uzir, J. Jovanović, and A. Pardo. Analytics of learning strategies: Associations with academic performance and feedback. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, LAK19, pages 461–470, New York, NY, USA, 2019. ACM.

[152] V. Mavroeidis and S. Bromander. Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 91–98. IEEE, 2017.

[153] M. May, S. George, and P. Prévot. Travis to enhance online tutoring and learning activities. *Interactive Technology and Smart Education*, 2011.

[154] S. McKenna, D. Staheli, and M. Meyer. Unlocking User-centered Design Methods for Building Cyber Security Visualizations. In *2015 IEEE Symp. on Vis. for Cyber Security (VizSec)*, pages 1–8. IEEE, 2015.

[155] M. E. McMurtrey, J. P. Downey, S. M. Zeltmann, and W. H. Friedman. Critical Skill Sets of Entry-level IT Professionals: An Empirical Examination of Perceptions from Field Personnel. *Journal of Information Technology Education: Research*, 7:101–120, 2008.

[156] MCR. The michigan cyber range. `https://www.merit.edu/cyberrange/`. Accessed: 2017-05-22.

[157] R. P. Medeiros, G. L. Ramalho, and T. P. Falcão. A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, 62(2):77–90, 2018.

[158] J. Medková, M. Husák, M. Vizváry, and P. Čeleda. Honeypot Testbed for Network Defence Strategy Evaluation. In *Proceedings of the 2017 IFIP/IEEE International Symposium on Integrated Network Management*, pages 887–888. IEEE Computer Society, 2017.

[159] K. E. Merrick. An Empirical Evaluation of Puzzle-based Learning as an Interest Approach for Teaching Introductory Computer Science. *IEEE Transactions on Education*, 53(4):677–680, 2010.

[160] D. R. Michael and S. L. Chen. *Serious Games: Games that Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.

[161] Z. Michalewicz and M. Michalewicz. Puzzle-based Learning. In *Proceedings of the 2007 AaeE Conference*, pages 1–8, 2007.

[162] Z. Michalewicz and M. Michalewicz. *Puzzle-based learning*. Hybrid Publishers, Ormond, Australia, 2008.

[163] D. Milojicic, I. M. Llorente, and R. S. Montero. OpenNebula A Cloud Management Tool. *IEEE INTERNET COMPUTING*, 15(2):11–14, MAR-APR 2011.

[164] J. Ministr and T. Pitner. Towards cybersecurity-qualified workforce. In *IDIMT-2019 Innovation and Transformation in a Digital World 27th Interdisciplinary Information Management Talks*. Trauner Verlag, Linz, Austria, 2019.

[165] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab. The DETER Project: Advancing the Science of Cyber Security Experimentation and Test. In *Proceedings of the 2010 IEEE International Conference on Technologies for Homeland Security (HST '10)*, Waltham, Massachusetts, Nov. 2010.

[166] J. Mirkovic and P. A. H. Peterson. Class capture-the-flag exercises. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, San Diego, CA, 2014. USENIX Association.

[167] S. Misra, V. Kumar, U. Kumar, K. Fantazy, and M. Akhter. Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, 29(9):972–980, 2012.

[168] J. Mokyr et al. *The gifts of Athena: Historical origins of the knowledge economy*. Princeton University Press, 2002.

[169] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen. Exploring Game Design for Cybersecurity Training. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*, pages 256–262. IEEE, 2012.

[170] NATO cooperative cyber defence centre of excellence. Locked shields. `http://ccdcoe.org/event/cyber-defence-exercises.html`. Accessed: 2017-05-22.

[171] NATO Supreme Headquarters Allied Powers Europe. Cyber coalition 16: Natos largest cyber defence exercise. `https://www.shape.nato.int/2016/cyber-coalition-16-ends-natos-largest-cyber-defence-exercise`, 2016. Accessed: 2017-08-28.

[172] NCR. The national cyber range. `http://www.acq.osd.mil/dte-trmc/docs/Docs/NCR/2015_NCR%20Info%20Sheet_Updated.pdf`. Accessed: 2017-05-22.

[173] D. M. Nicol, W. H. Sanders, and K. S. Trivedi. Model-based evaluation: from dependability to security. *IEEE Transactions on dependable and secure computing*, 1(1):48–65, 2004.

[174] D. A. Norman and S. W. Draper. *User Centered System Design: New Perspectives on Human-computer Interaction*. CRC Press, 1986.

[175] J. Olsson and M. Boldt. Computer forensic timeline visualization tool. *Digital Investigation*, 6:S78 – S87, 2009. The Proceedings of the Ninth Annual DFRWS Conference.

[176] T. Opsahl, F. Agneessens, and J. Skvoretz. Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths. *Social networks*, 32(3):245–251, 2010.

[177] R. Ošlejšek and T. Pitner. Optimization of cyber defense exercises using balanced software development methodology. *International Journal of Information Technologies and Systems Approach.*, 14(1), 2021.

[178] R. Ošlejšek, V. Rusňák, K. Burská, V. Švábenský, J. Vykopal, and J. Čegan. Conceptual model of visual analytics for hands-on cybersecurity training. *IEEE Transactions on Visualization and Computer Graphics*, 27(8):3425–3437, 2021.

[179] R. Ošlejšek, V. Rusňák, K. Burská, V. Švábenský, and J. Vykopal. Visual feedback for players of multi-level capture the flag games: Field usability study. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–11. IEEE Computer Society, 2019.

[180] V. E. Owen and R. S. Baker. Fueling prediction of player decisions: Foundations of feature engineering for optimized behavior modeling in serious games. *Technology, Knowledge and Learning*, 25(2):225–250, 2020.

[181] R. Ošlejšek, D. Toth, Z. Eichler, and K. Burská. Towards a unified data storage and generic visualizations in cyber ranges. In N.-A. L.-K. Mark Scanlon, editor, *Proceedings of the 16th European Conference on Cyber Warfare and Security ECCWS 2017*, pages 298–306, UK, 2017. Academic Conferences and Publishing International Limited.

[182] R. Ošlejšek, J. Vykopal, K. Burská, and V. Rusňák. Evaluation of cyber defense exercises using visual analytics process. In *2018 IEEE Frontiers in Education Conference*, pages 1–9, San Jose, California, USA, 2018. IEEE.

[183] A. Perez-Garcia, C. Siaterlis, and M. Masera. Designing repeatable experiments on an emulab testbed. In *International Conference on Broadband Communications, Networks and Systems*, pages 28–39. Springer, 2010.

[184] G. Petty. *Teaching Today: A Practical Guide*. Nelson Thornes, 2009.

[185] W. M. Petullo, K. Moses, B. Klimkowski, R. Hand, and K. Olson. The Use of Cyber-Defense Exercises in Undergraduate Computing Education. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, Austin, TX, 2016. USENIX Association.

[186] C. Pham, D. Tang, K.-i. Chinen, and R. Beuran. Cyris: A cyber range instantiation system for facilitating security training. In *Proceedings of the Seventh Symposium on Information and Communication Technology*, SoICT '16, pages 251–258, New York, NY, USA, 2016. ACM.

[187] A. Pras, A. Sperotto, G. Moura, I. Drago, R. Barbosa, R. Sadre, R. Schmidt, and R. Hofstede. Attacks by "anonymous" wikileaks proponents not anonymous. Technical report, University of Twente, Centre for Telematics and Information Technology (CTIT), 2010.

[188] Questionmark Computing Limited. Questionmark perception. `https://www.questionmark.com`. (Accessed on 2018-06-13).

[189] A. S. Raj, B. Alangot, S. Prabhu, and K. Achuthan. Scalable and lightweight ctf infrastructures using application containers. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, Austin, TX, Aug. 2016. USENIX Association.

[190] Red Hat. Ansible. `https://www.ansible.com`. Accessed: 2017-05-22.

[191] D. Restuccia. Job Market Intelligence: Cybersecurity Jobs. Technical report, Burning Glass Tech, 2015.

[192] J. C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Fifth Int. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71. IEEE, 2007.

[193] M. Rodríguez-Triana, L. Prieto, et al. Monitoring, Awareness and Reflection in Blended Technology Enhanced Learning: a Systematic Review. *International Journal of Technology Enhanced Learning*, 9, 02 2016.

[194] L. Rossey. SimSpace cyber range. `https://www.acsac.org/2015/program/ACSAC%202015%20CEF%20Panel%20-%20Rossey.pdf`. ACSAC 2015 Panel: Cyber Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cybersecurity Research.

[195] N. Rowe and S. Garfinkel. Finding Anomalous and Suspicious Files from Directory Metadata on a Large Corpus. In *Proceedings of the Digital Forensics and Cyber Crime*, 2011.

[196] J. Rubin. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[197] I. Ruiz-Rube, J. M. Dodero, M. Palomo-Duarte, M. Ruiz, and D. Gawn. Uses and applications of software & systems process engineering meta-model process models. a systematic mapping study. *Journal of Software: Evolution and Process*, 25(9):999–1025, 2013.

[198] S. Ryan and R. V. O'Connor. Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, 55(9):1614–1624, 2013.

[199] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, E. G., and D. A. Keim. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, Dec 2014.

[200] J. Saldaña. *The Coding Manual for Qualitative Researchers*. Sage, 2015.

[201] SANS Institute. NetWars: DFIR Tournament. (Accessed on 2018-06-13).

[202] J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform, 2011.

[203] J. Sauro and J. S. Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1599–1608, New York, NY, USA, 2009. ACM.

[204] W. Schepens, D. Ragsdale, J. R. Surdu, J. Schafer, and R. New Port. The cyber defense exercise: An evaluation of the effectiveness of information assurance education. *The Journal of Information Security*, 1(2), 2002.

[205] M. Scherr. Multiple and coordinated views in information visualization. *Trends in Information Visualization*, 38:1–33, 2008.

[206] G. Schneider and J. P. Winters. *Applying Use Cases: A Practical Guide*. Pearson Education, 2001.

[207] S. Schwab, B. Wilson, C. Ko, and A. Hussain. SEER: A security experimentation environment for DETER. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007*. USENIX Association, 2007.

[208] D. Schweitzer and W. Brown. Using Visualization to Teach Security. *Journal of Computing Sciences in Colleges*, 24(5):143–150, 2009.

[209] B. A. Schwendimann, M. J. Rodríguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet, and P. Dillenbourg. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, 10(1):30–41, Jan 2017.

[210] Security Competence. Hacking-lab. `http://www.hacking-lab-ctf.com/technical.html`. Accessed: 2017-05-22.

[211] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. on Vis. and Computer Graphics*, 18(12):2431–2440, 2012.

[212] E. Seker and H. H. Ozbenli. The concept of cyber defence exercises (cdx): Planning, execution, evaluation. In *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–9. IEEE, 2018.

[213] Y. Shi, Y. Liu, H. Tong, J. He, G. Yan, and N. Cao. Visual analytics of anomalous user behaviors: A survey. *arXiv preprint arXiv:1905.06720*, 2019.

[214] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[215] A. K. Shuja and J. Krebs. *IBM Rational unified process reference and certification guide: solution designer (RUP)*. Pearson Education, 2007.

[216] C. Siaterlis, A. P. Garcia, and B. Genge. On the Use of Emulab Testbeds for Scientifically Rigorous Experiments. *IEEE Communications Surveys Tutorials*, 15(2):929–942, Second 2013.

[217] G. Siemens and P. Long. Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, 46(5):30, 2011.

[218] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu. AVOIDIT: A cyber attack taxonomy. In *9th Annual Symposium on Information Assurance (ASIA'14)*, pages 2–12, 2014.

[219] U. J. Staff. Joint Training Manual for the Armed Forces of the United States (CJCSM 3500.03 D). *Washington, DC: Joint Chiefs of Staff*, 2012.

[220] D. Staheli, T. Yu, R. J. Crouser, S. Damodaran, K. Nam, D. O'Gwynn, S. McKenna, and L. Harrison. Visualization Evaluation for Cyber Security: Trends and Future Directions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 49–56. ACM, 2014.

[221] J.-E. Stange, M. Dörk, J. Landstorfer, and R. Wettach. Visual filter: graphical exploration of network security log files. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 41–48, 2014.

[222] K. E. Stewart, J. W. Humphries, and T. R. Andel. Developing a Virtualization Platform for Courses in Networking, Systems Administration and Cyber Security Education. In *Proceedings of the 2009 Spring Simulation Multiconference*, page 65. Society for Computer Simulation International, 2009.

[223] M. Suby and F. Dickson. The 2013 (isc) 2 global information security workforce study. *Frost & Sullivan in partnership with Booz Allen Hamilton for ISC2*, 2013.

[224] V. Sébastien, D. Sébastien, I. Timol, D. Gay, A. Cucchi, and C. Porlier. Moodleboard: Dynamic and Interactive Indicators for Teachers and Pedagogical Engineers. In *2019 Conf. on Next Generation Computing Applications (NextComp)*, pages 1–5, New York, Sep. 2019. IEEE.

[225] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *2012 IEEE Pacific Visualization Symposium*, pages 41–48. IEEE, 2012.

[226] D. R. Thomas. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, 27(2):237–246, 2006.

[227] D. Tovarňák and T. Pitner. Continuous queries over distributed streams of heterogeneous monitoring data in cloud datacenters. In *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, pages 470–481, Aug 2014.

[228] D. Turk, R. France, and B. Rumpe. Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*, 2014.

[229] A. Ulmer, D. Sessler, and J. Kohlhammer. Netcapvis: Web-based progressive visual analytics for network packet captures. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2019.

[230] C. Vaitsis, G. Nilsson, and N. Zary. Big data in medical informatics: improving education through visual analytics. In *MIE*, pages 1163–1167, 2014.

[231] B. Van Leeuwen, V. Urias, J. Eldridge, C. Villamarin, and R. Olsberg. Performing cyber security analysis using a live, virtual, and constructive (LVC) testbed. In *Military Communications Conference 2010 - MILCOM 2010*, pages 1806–1811, 2010.

[232] R. Vatrapu, C. Teplovs, N. Fujita, and S. Bull. Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pages 93–98. ACM, 2011.

[233] P. Velan and R. Krejčí. Flow information storage assessment using ipfixcol. In *Dependable Networks and Services*, volume 7279 of *Lecture Notes in Computer Science*, pages 155–158. Springer Berlin Heidelberg, 2012.

[234] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos. Learning Analytics Dashboard Applications. *Am. Behavioral Scientist*, 57(10):1500–1509, 2013.

[235] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx. Learning Dashboards: An Overview and Future Research Opportunities. *Personal and Ubiquitous Computing*, 18:1499–1514, 2013.

[236] G. Vigna, K. Borgolte, J. Corbetta, A. Doupe, Y. Fratantonio, L. Invernizzi, D. Kirat, and Y. Shoshitaishvili. Ten years of iCTF: The good, the bad, and the ugly. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[237] J. Voříšek, J. Pour, and A. Buchalcevová. Management of business informatics model: principles and practices. *Ekonomie a Management*, 18(3):160–173, 2015.

[238] P. Vorobkalov and V. Kamaev. Quality Estimation of e-Learning Systems. In *Third International Conference "Modern (e-) Learning" MeL 2008*. Inst. of Information Theories and Applications FOI ITHEA, 2008.

[239] V. Švábenský, J. Vykopal, M. Čermák, and M. Laštovička. Enhancing Cybersecurity Skills by Creating Serious Games. In *Proc. of the 23rd Annual sConf. on Innovation and Technology in Computer Science Education*, pages 194–199. ACM, 2018.

[240] J. Vykopal and M. Barták. On the Design of Security Games: From Frustrating to Engaging Learning. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, Austin, TX, Aug. 2016. USENIX Association.

[241] J. Vykopal and J. Cegan. Finding Exercise Equilibrium: How to Support the Game Balance at the Very Beginning? In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, pages 719–719, New York, NY, USA, 2017. ACM.

[242] J. Vykopal, R. Ošlejšek, K. Burská, and K. Zákopčanová. Timely feedback in unstructured cybersecurity exercises. In *Proceedings of Special Interest Group on Computer Science Education, Baltimore, Maryland, USA, February 21–24, 2018(SIGCSE'18)*, pages 173–178, Baltimore, Maryland, USA, 2018. ACM.

[243] J. Vykopal, R. Ošlejšek, P. Čeleda, M. Vizváry, and D. Tovarňák. Kypo cyber range: Design and use cases. In *Proceedings of the 12th International Conference on Software Technologies - Volume 1: ICSOFT*, pages 310–321, Madrid, Spain, 2017. SciTePress.

[244] J. Vykopal, M. Vizváry, R. Ošlejšek, P. Čeleda, and D. Tovarňák. Lessons learned from complex hands-on defence exercises in a cyber range. In *2017 IEEE Frontiers in Education Conference*, pages 1–8, Indianapolis, IN, USA, 2017. IEEE.

[245] J. Vykopal, V. Švábenský, and E. Chang. Benefits and pitfalls of using capture the flag games in university courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*, pages 752–758, Portland, Oregon, USA, 2020. ACM.

[246] M. Wang, S. Carr, J. Mayo, C.-K. Shene, and C. Wang. Mlsvisual: A visualization tool for teaching access control using multi-level security. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, page 93–98, New York, NY, USA, 2014. Association for Computing Machinery.

[247] M. Wang, J. Mayo, C.-K. Shene, T. Lake, S. Carr, and C. Wang. Rbacvisual: A visualization tool for teaching access control using role-based access control. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '15, page 141–146, New York, NY, USA, 2015. Association for Computing Machinery.

[248] A. Weijters and W. M. van der Aalst. Process Mining: Discovering Workflow Models from Event-based Data. In *Belgium-Netherlands Conf. on Artificial Intelligence*. Citeseer, 2001.

[249] R. Weiss, F. Turbak, J. Mache, and M. E. Locasto. Cybersecurity education and assessment in edurange. *IEEE Security & Privacy*, 15(3):90–95, 2017.

[250] D. Welch, D. Ragsdale, and W. Schepens. Training for information assurance. *Computer*, 35(4):30–37, Apr 2002.

[251] J. Werther, M. Zhivich, T. Leek, and N. Zeldovich. Experiences in Cyber Security Education: The MIT Lincoln Laboratory Capture-the-flag Exercise. In *Proceedings of*

the 4th Conference on Cyber Security Experimentation and Test, CSET'11. USENIX Association, 2011.

[252] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):255–270, Dec. 2002.

[253] P. C. Wong and J. Thomas. Guest editors' introduction–visual analytics. *IEEE Computer Graphics and Applications, 24 (5): 20-21*, 24(PNNL-SA-41935), 2004.

[254] P. C. Wong and J. Thomas. Visual Analytics. *IEEE Computer Graphics and Applications*, (5):20–21, 2004.

[255] J. Wroclawski, J. Mirkovic, T. Faber, and S. Schwab. A two-constraint approach to risky cybersecurity experiment management. In *Sarnoff Symposium*, 2008. Invited paper.

[256] M. M. Yamin and B. Katt. Inefficiencies in cyber-security exercises life-cycle: A position paper. In *AAAI Fall Symposium: ALEC*, pages 41–43, 2018.

[257] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1633–1636, New York, NY, USA, 2010. ACM.

[258] Y. Yoneyama, K. Matsushita, K. J. Mackin, M. Ohshiro, K. Yamasaki, and E. Nunohiro. Puzzle Based Programming Learning Support System with Learning History Management. In *Proceedings of the 16th International conference on Computers in Education*, pages 623–627, 2008.

[259] X. Yuan, P. Vega, Y. Qadah, R. Archer, H. Yu, and J. Xu. Visualization Tools for Teaching Computer Security. *ACM Transactions on Computing Education (TOCE)*, 9(4):20, 2010.

[260] J. Yuen, B. Turnbull, and J. Hernandez. Visual Analytics for Cyber Red Teaming. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8. IEEE, 2015.

[261] K. M. Zaki and R. Moawad. A hybrid disciplined agile software process model. In *2010 The 7th International Conference on Informatics and Systems (INFOS)*, pages 1–8. IEEE, 2010.

[262] K. Zhang, S. Dong, G. Zhu, D. Corporon, T. McMullan, and S. Barrera. picoctf 2013-toaster wars: When interactive storytelling game meets the largest computer security competition. In *Games Innovation Conference (IGIC), 2013 IEEE International*, pages 293–299. IEEE, 2013.

[263] P. Čeleda, J. Čegan, J. Vykopal, and D. Tovarňák. KYPO – A Platform for Cyber Defence Exercises. In *STO-MP-MSG-133: M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence*, page 12. NATO Science and Technology Organization, 2015.

[264] S. Špaček, P. Čeleda, M. Drašar, and M. Vizváry. Analyzing an Off-the-Shelf Surveillance Software: Hacking Team Case Study. *Security and Protection of Information*, 2017.