Enhancing Situational Awareness for Tutors of Cybersecurity Capture the Flag Games

1st Karolína Dočkalová Burská
Faculty of Informatics
Masaryk University
Brno, Czech Republic
xburska@fi.muni.cz

2nd Vít Rusňák
Institute of Computer Science
Masaryk University
Brno, Czech Republic
rusnak@ics.muni.cz

3rd Radek Ošlejšek
Faculty of Informatics
Masaryk University
Brno, Czech Republic
oslejsek@fi.muni.cz

***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.**

***Index Terms*—cybersecurity education, hands-on training, situational awareness**

I. 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 [21], [22]. In the cybersecurity domain, hands-on learning is primarily represented by *Capture the Flag* (CTF) games [7], [31], [36]. Michalewicz et al. [24] 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 [14], [23], [37]. Multiple studies confirm the usefulness of puzzle-based learning also for cybersecurity education [6], [11], [15].

Hands-on cybersecurity training is often organized in so-called cyber ranges [4], [8], [35]. The data and field observations referenced in this paper were obtained during the training sessions in *KYPO Cyber Range*[1] [38]. 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 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

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) and REWIRE "Sector Skills Alliances 2020" (No. 621701-EPP-1-2020-1-LT-EPPKA2-SSA-B). 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).

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

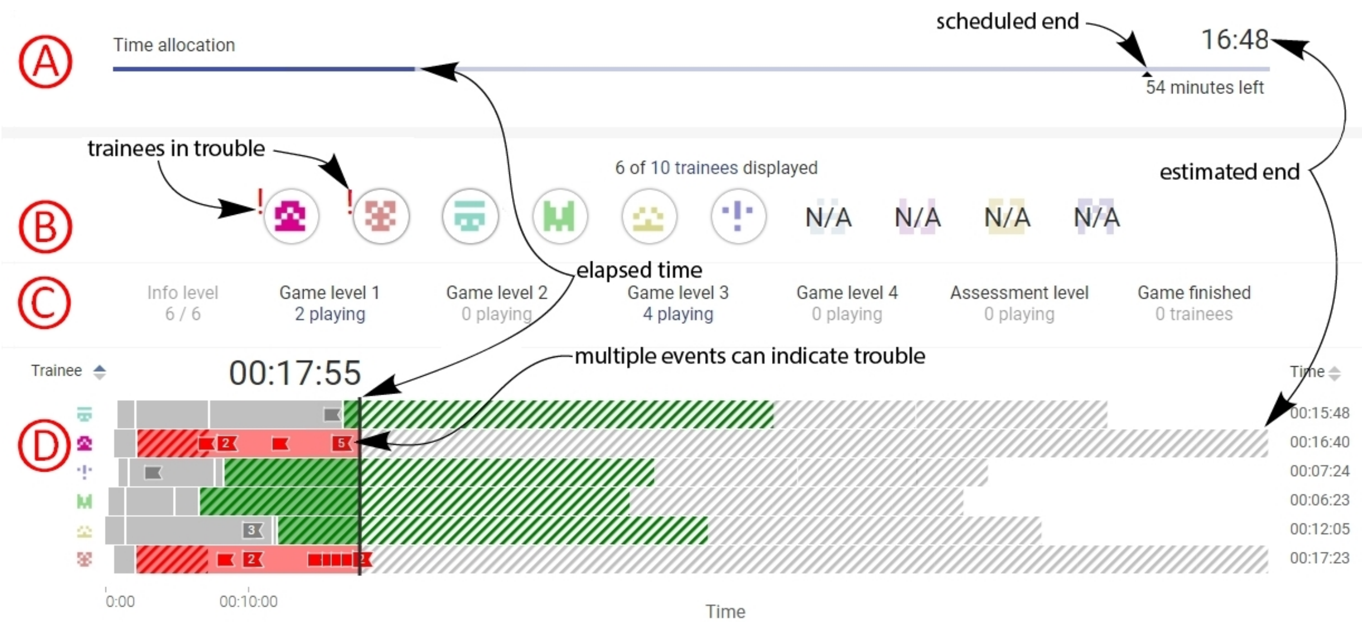

Fig. 1. Progress Visualization Tool (PVT) serves as a visual overview of ongoing hands-on training session. The tutors can quickly identify outstanding situations that may require their intervention. The tool consists of four sections which provide complementary information: A – Timeline, B – Trainees, C – Game Level Occupancy, and D – Detailed Timeline.

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.

II. Related Work

According to the classification provided by Oslejsek et al. [25], 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 [27]. Also, Macfadyen and Dawson [19] 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. [12], [13] proposed a general-purpose web-based environment for the visualization of Moodle activities to increase awareness and support self-reflection. Deeb and Hickey [9] 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. [17] 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. [16] 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 [20]. 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 [18] or access control models [33], [34], 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 [2] 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.

III. Functional Requirements and Data Abstraction

We design the tool iteratively, guided by the design study methodology framework [30]. 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.

A. 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.

B. 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.

IV. 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 1: timeline, trainees, game level occupancy, and detailed timeline). From top to bottom, each level adds more details to the upper ones.

A. Timeline

The timeline section (Figure 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.

B. Trainees

The interactive list of trainees (Figure 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).

C. Game Level Occupancy

Arranged in a horizontal list, the game level occupancy section (Figure 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.

D. 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 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.

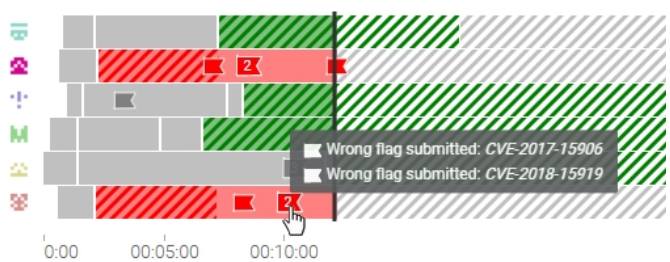

Fig. 2. A detailed timeline with aggregated events and details shown on demand.

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 2) 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.

V. Design Decisions

When designing the PVT, we put emphasis on using the situation awareness design principles specified by Endsley [10]. 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.

VI. 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.

A. 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).

B. 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 [28] and the SEQ – Simple Ease Question [29], 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 [29].

C. 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[2] training scenario data (description of the tasks, scoring, etc.) and events generated by trainees as described in Section III-B. DS1 and DS2 cannot be published due to content protection policies.

D. 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 [32] 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 [3]). Moreover, low

[2]We replaced hints and solutions with dummy texts and modified correct flags.

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 [28]. 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 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.

VII. 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.

A. 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.

B. 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 [1], [5], [26]. 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.

C. 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.

VIII. 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.

References

[1] 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.

[2] 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.

[3] A. Bangor, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies*, 4(3):114–123, May 2009.

[4] 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.

[5] 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.

[6] 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.

[7] 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)*, pages 1–9, San Diego, CA, 2014. USENIX Association.

[8] J. Davis and S. Magrath. A survey of cyber ranges and testbeds. Technical report, DTIC Document, 2013.

[9] F. A. Deeb and T. Hickey. Classroom Orchestration with Problem Solving Markov Models. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, New York, Oct 2018. IEEE Press.

[10] M. R. Endsley. *Designing for situation awareness: An approach to user-centered design*. CRC press, Boca Raton, USA, 2016.

[11] M. Gondree, Z. N. Peterson, and T. Denning. Security Through Play. *IEEE Security & Privacy*, 11(3):64–67, 2013.

[12] 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.

[13] 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, Berlin, Heidelberg, 2010. Springer, Springer Berlin Heidelberg.

[14] 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, New York, 2015. IEEE, IEEE Press.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] M. May, S. George, and P. Prévot. Travis to enhance online tutoring and learning activities. *Interactive Technology and Smart Education*, 2011.

[21] 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 Inf. Technology Education: Research*, 7:101–120, 2008.

[22] 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.

[23] K. E. Merrick. An Empirical Evaluation of Puzzle-based Learning as an Interest Approach for Teaching Introductory Computer Science. *IEEE Trans. on Education*, 53(4):677–680, 2010.

[24] Z. Michalewicz and M. Michalewicz. *Puzzle-based learning*. Hybrid Publishers, Ormond, Australia, 2008.

[25] R. Ošlejšek, V. Rusňák, K. Burská, V. Švábenský, J. Vykopal, and J. Cegan. Conceptual model of visual analytics for hands-on cybersecurity training. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–13, 2020.

[26] 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.

[27] 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.

[28] J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform, USA, 2011.

[29] 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.

[30] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012.

[31] V. Švábenský, J. Vykopal, M. Cermak, and M. Laštovička. Enhancing Cybersecurity Skills by Creating Serious Games. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 194–199, New York, NY, USA, 2018. ACM, ACM.

[32] D. R. Thomas. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, 27(2):237–246, 2006.

[33] 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.

[34] 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.

[35] R. Weiss, F. Turbak, J. Mache, and M. E. Locasto. Cybersecurity education and assessment in edurange. *IEEE Security & Privacy*, 15(3):90–95, 2017.

[36] 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, pages 1–12, Berkeley, CA, USA, 2011. USENIX Association.

[37] 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, New York, 2008. IEEE Press.

[38] 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, Munich (Germany), 2015. NATO Science and Technology Organization.