

Reinforcing Cybersecurity Hands-on Training With Adaptive Learning

Pavel Seda
*Faculty of Informatics
Masaryk University
Brno, Czech Republic
seda@fi.muni.cz*

Jan Vykopal
*Faculty of Informatics
Masaryk University
Brno, Czech Republic
vykopal@fi.muni.cz*

Valdemar Švábenský
*Faculty of Informatics
Masaryk University
Brno, Czech Republic
svabensky@fi.muni.cz*

Pavel Čeleda
*Faculty of Informatics
Masaryk University
Brno, Czech Republic
celeda@fi.muni.cz*

Abstract—This Research To Practice Full Paper presents how learning experience influences students’ capability to learn and their motivation for further learning. Although each student is different, standard instruction methods do not adapt to individual students. Adaptive learning reverses this practice and attempts to improve the student experience. While adaptive learning is well-established in programming, it is rarely used in cybersecurity education. This paper is one of the first works investigating adaptive learning in cybersecurity training. First, we analyze the performance of 95 students in 12 training sessions to understand the limitations of the current training practice. Less than half of the students (45 out of 95) completed the training without displaying any solution, and only in two sessions, all students completed all phases. Then, we simulate how students would proceed in one of the past training sessions if it would offer more paths of various difficulty. Based on this simulation, we propose a novel tutor model for adaptive training, which considers students’ proficiency before and during an ongoing training session. The proficiency is assessed using a pre-training questionnaire and various in-training metrics. Finally, we conduct a case study with 24 students and new training using the proposed tutor model and adaptive training format. The results show that the adaptive training does not overwhelm students as the original static training format. In particular, adaptive training enables students to enter several alternative training phases with lower difficulty than the phases in the original training. The proposed adaptive format is not restricted to particular training used in our case study. Therefore, it can be applied to practicing any cybersecurity topic or even in other related computing fields, such as networking or operating systems. Our study indicates that adaptive learning is a promising approach for improving the student experience in cybersecurity education. We also highlight diverse implications for educational practice that improve students’ experience.

Index Terms—adaptive learning, case study, cybersecurity, evaluation, tutor model

I. INTRODUCTION

Learning cybersecurity requires extensive knowledge and skills, ranging from a wide area of theoretical concepts to practical skills with operating systems, command-line tools, and system vulnerabilities [1]. As a result, it is difficult to conduct hands-on cybersecurity training that would match the skills of all participants in the training. This situation is further complicated since more and more students with different backgrounds are entering the field of cybersecurity [2].

Although the instructor can intervene to help students interactively, this is feasible only in relatively small classes, and

not every student actively asks for help. The interactive help is especially complicated during online training (e.g., forced by restrictions caused by the COVID-19 pandemic [3]).

To support our assumptions that students do not fully benefit from the training sessions, we analyze 12 hands-on training sessions on various cybersecurity topics we held in 2019 and 2020. We observed that only 47% of students successfully completed the training (for more information, see Section III-A).

We see the opportunity to improve the students’ experience and skills using an intelligent tutoring system (ITS), which adapts the learning environment according to the student’s abilities. Unfortunately, an ITS in the domain of hands-on cybersecurity training is rare, mostly because the interactive lab environment and its setup differ for particular sessions. As a result, cybersecurity platforms offer static scenarios with limited or no adaptiveness [4]. We could create an ITS for a specific training session. This would bring great flexibility in defining the conditions for serving adaptive tasks to students. However, such ITS could not be reused for another training. Our main goal is to create a concept of generic cybersecurity training that will adapt to the current phase of individual student skills.

In this paper, we present a generic format for adaptive training and a tutor model. The model determines appropriate tasks based on students’ theoretical knowledge and current performance. Using the proposed format and model, we conduct a case study involving cybersecurity hands-on training with 24 undergraduate students and graduates in computer science. We report teaching experience from the execution of adaptive hands-on training based on the proposed tutor model implemented in KYPO Cyber Range Platform (CRP) [5]. The results suggest that adaptive training increases the chances of successful completion of training and deepens the experience and knowledge gained from the training. In our study, 88% of students completed the training without asking for a solution of any task. Further, most of the students reported that they did not get stuck at any point of the training and enjoyed it. Finally, we provide recommendations for instructors on using the proposed format and model and also depict future research directions.

This paper is organized into six sections. Section II pro-

vides an overview of ITSs in computer science education. Section III describes our past experience and motivation. Section IV details the training format and the newly developed tutor model. Section V describes case study setup, including teaching context and participants. Section VI reports the results from three hands-on training sessions. Finally, Section VII concludes the paper and outlines future work.

II. RELATED WORK

Adaptive learning techniques are a well-established research area [6] that accommodates the pedagogical content for the learners and their current state of knowledge. These techniques were introduced in the 1970s [7], and the research area still receives considerable interest. Personalized learning achievable by adaptive techniques was identified by the US National Academy of Engineering as one of the Grand Challenges for Engineering [8].

We start with ITS that conceptualize adaptive learning in a way that is commonly accepted in computer science. ITS typically contain the following parts: (i) *domain model*, (ii) *student model*, (iii) *tutor model*, and (iv) *user interface model* [9]. The *domain model* presents educational content and its relationships [9]. The *student model* captures the students' knowledge to assess their performance [10], [11]. The *tutor model (instructional policy)* presents the suitable learning tasks to students [12]. Finally, a *user interface model* interacts with the user via a pre-defined interface [9, chapter 9].

Although the ITS research area is well established, to the best of our knowledge, there are no available ITS models for comprehensive hands-on cybersecurity training in a networked lab environment. For that reason, we review the ITS research from other domains, which improve or discuss student models to evaluate the participants' performance and tutor models to assign suitable tasks. Effenberger and Pelánek [13] discuss several approaches to measure the student's performance during introductory programming tasks. They find that the widely used performance measure called binary success is not suitable for the evaluation of programming tasks since it contains too little information. The evaluation of programming tasks is harder than the evaluation of answering multiple-choice questions about any topic in computer science. Therefore, they propose multiple qualitative and quantitative methods, based on the four performance levels *failed*, *poor*, *good*, and *excellent*. Khosravi et. al [14] provide lessons learned from using the Ripple system that recommends suitable learning activities for students of relational databases. The authors found that an important part of the learning system is based on gamification, such as awards and leaderboards to motivate students. Further, [15] uses Bloom's taxonomy to dynamically adapt the training process. The authors define several layers with different difficulties that should be accomplished. The system evaluates the students' exercises and exams during the training. After reaching a good understanding, the student can proceed with a related advanced training scenario. Contrary to our approach, it seems that their adaptiveness is mostly based on exam scores and does not include more detailed

metrics such as the commands used in an interactive learning environment. For more information on ITS, we suggest [9] that focuses on design recommendations and [6], [16], [17] that review the recent research.

Next, the participants' perceptions of difficulty are subjective. Nebel et. al [18] discussed that perceived difficulty within a competition might differ relative to each learner's performance. A participant winning effortlessly might indicate a low difficulty, whereas a losing participant may perceive a relatively high difficulty even if the context is identical. This argumentation appears evident but is important. The individual difficulty might play a crucial role in influencing the students' experience and how the learning process evolves. Xue et al. [19] observed that perceiving the difficulty influences participant engagement and how often the training is played.

Finally, we searched for related works in the area of cybersecurity education. We found only a few relevant sources about adaptiveness in cybersecurity training. Hatzivasilis et al. [15] propose suitable assignments of cybersecurity tasks to students in exercises held in a cyber range. However, they do not propose the unified design of adaptive hands-on training. In the industry sector, the Circadence company provides adaptable cyber training and learning opportunities. However, their platform does not support adaptive task assignments based on the students' performance and mainly focuses on the adaptive pre-configuration of training sessions. This includes turning the hints and chatbot on or off during the training [20].

Based on the available literature and eight years of our experience with hands-on cybersecurity training, we believe the reason for the absence of ITS in comprehensive hands-on cybersecurity education is the high complexity of systems (hardware, software, and domain knowledge requirements).

III. OUR TEACHING EXPERIENCE AND EXPECTATIONS FROM ADAPTIVE LEARNING

This section presents our previous experience with non-adaptive training sessions and our expectations from introducing adaptivity to hands-on training.

A. Our Teaching Experience

We have been designing and organizing cybersecurity training sessions since 2014 [21]. The participating high-school and undergraduate students, as well as professional learners, value the hands-on nature of these sessions and the opportunities to practice cyber attacks and defense. On the other hand, many participants were frustrated in various phases of the training, even though it contained on-demand hints. The participants lacked some prerequisite skills and knowledge or wanted to complete the training without help.

To validate our assumptions about the factors influencing students' learning experience, we analyzed interaction data from 12 training sessions held in 2019 and 2020. The data were collected automatically in the KYPO CRP [5]. A total of 95 students participated in one of 12 cybersecurity training sessions. Each training comprised three to six consecutive

phases. In total, less than half of the participants (45 out of 95) completed their training sessions, i.e., completed all phases without displaying any solution. In two training sessions, all participants completed all phases. In the other ten sessions, the ratio of successful participants ranged from 0 to 83% (median 55%). The count of phases that participants completed in the same training session varied too.

These student difficulties can be mitigated by conducting training sessions that adapt to the proficiency and current progress of each student. However, conducting such adaptive training sessions is infeasible without a training tutor integrated into the platform. To support this argument, we counted the actions the students performed in the previous training sessions (see Table I). These actions include starting the training phase, submitting the correct or incorrect answer in a phase, and displaying a hint or solution. All these actions are automatically processed by the tutor without instructor intervention. In the analyzed training sessions, the average number of actions per participant ranged from 17 to 62 (median 29). This number is too high to conduct the training sessions manually (by the instructor) without the support of the software in the learning environment.

TABLE I
STATISTICS OF THE PAST NON-ADAPTIVE TRAINING SESSIONS.

Training session	Completed by participants [%]	Most participants ended in phase	Avg actions per participant
1	100	3 out of 3	28
2	100	4 out of 4	21
3	0	2 out of 4	29
4	83	2 out of 5	29
5	60	3 out of 5	25
6	0	3 out of 5	40
7	57	1 out of 5	17
8	25	1 out of 5	45
9	18	3 out of 6	62
10	52	3 out of 6	29
11	33	2 out of 5	35
12	66	5 out of 5	30

B. Adaptive Learning Expectations

Our initial assumption for the integration of adaptivity to the training was that fewer students will fail the training. Further, we suppose they finish the training to the best of their capability and thus fully benefit from the training.

Since adaptive learning was not used in the previous cybersecurity hands-on training, we simulated how students would proceed in one of our previous training sessions, which we made adaptive to students' proficiency and performance. We chose a training with six phases including (i) network reconnaissance using `nmap`, (ii) finding a vulnerability, (iii) exploiting the vulnerability using `Metasploit`, (iv) Linux operations, (v) cracking a SSH passphrase, and (vi) connecting via SSH using the cracked passphrase and displaying the content of the file.

In our simulation, the adaptivity of the training lies in modifying the difficulty of the tasks presented to each student in all six training phases. We created two new tasks for each

phase that contains one or more hints in the assignment to simplify the phase. Next, we selected the metrics gathered in the KYPO CRP. The metrics used for our simulation were: (i) *pre-training assessment*, (ii) *training completion time*, and (iii) *actions in the learning environment including entered commands*. The pre-training assessment is a questionnaire before the training that maps the theoretical knowledge and self-assessment of skills of the participants relevant to the training. The training completion time captures how long the participant solved a training phase. The actions in the learning environment are commands entered in the learning environment during the training, submissions of the wrong answers, or displaying the solution of the task. In particular, we count a number of entered commands relevant to a particular phase. For instance, too many entered `ssh` commands may indicate that a participant lacks skills in using this particular command. We employed these metrics to find the most suitable task in each phase for the participant, as shown in Table II.

TABLE II
METRICS USED FOR DETERMINING THE MOST SUITABLE TASK
(✓ = METRIC USED, ✗ = METRIC NOT USED).

Training phase	Pre-training assessment	Performance	Actions
1	✓	✗	✗
2	✓	✓	✓
3	✓	✓	✗
4	✓	✓	✗
5	✓	✓	✗
6	✗	✓	✓

We developed simulation software that processes the data from the non-adaptive training session to calculate the transitions of participants between variant tasks based on the described metrics. The simulated transitions of 23 participants are shown in a Sankey chart in Figure 1. The original, non-adaptive training consisted of six tasks: P1T1, P2T1, P3T1, P4T1, P5T1, and P6T1. The newly added, alternative tasks are those denoted T2 or T3, i.e., P1T2, P1T3, P2T2, P2T3, etc.

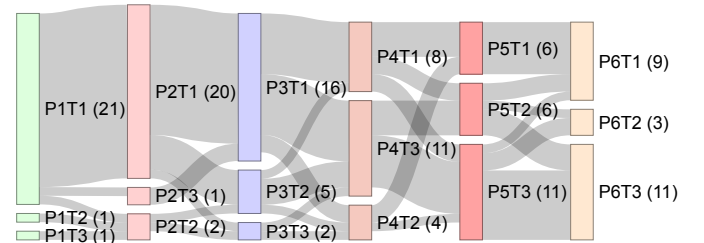


Fig. 1. Transitions of participants between particular tasks in training. PXTY denotes task TY in the phase PX. The number of participants who attempted to solve the task is in brackets.

We see the participants would enter not only the original tasks (T1) but also new easier variant tasks (T2 or T3), which indicates the adaptive training would be beneficial for our diversely performing participants. In particular, 17 out of 23 participants would benefit from this adaptive training because

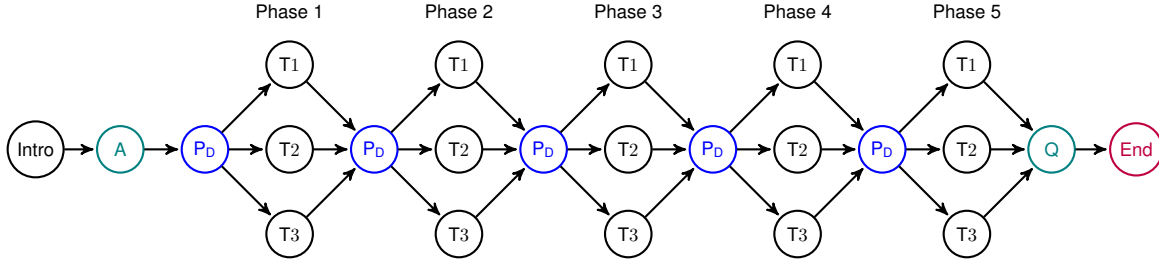


Fig. 2. Graph structure of adaptive cybersecurity training with pre-training assessment (A), decision component (P_D) applying the proposed model, and a post-training questionnaire (Q). This exemplary training contains five phases. Each phase contains one base task (T1) and two variant tasks (T2, T3).

they would get one or more variant tasks matching their skills better. These results strengthen our expectation that adaptive learning techniques may increase the students' experience and reduce the number of students that get stuck during the hands-on training.

Nevertheless, the software was specifically developed for one training and does not provide a generic solution for cybersecurity training with different topics in phases and different relations between its phases. We address this limitation in the next section.

IV. DESIGN OF ADAPTIVE CYBERSECURITY TRAINING

In this section, we present a generic format of adaptive cybersecurity hands-on training based on a model that uses the students' knowledge and performance to assign suitable training tasks. We evaluate the format using a case study presented in Section V.

A. Training Format

We propose a generic structure for adaptive cybersecurity training. Figure 2 shows an example of such structure with five phases, each with three tasks of various difficulty. In general, the training can contain an arbitrary number of phases and tasks. The training consists of several components: the introduction (Intro), the pre-training assessment (A), training phases including variant tasks (TX), decision components (P_D), and post-training questionnaire (Q).

First, the introduction familiarizes the student with the training and communicates all necessary information before the training start.

The pre-training assessment is the first component collecting data about students' knowledge and skills. The questions asked in the pre-training assessment are grouped into the *question groups* by their relation to specific training phases. Each question can be assigned into several question groups since they can be relevant to more phases. For each training phase, we set the *essential ratio* of knowledge to determine whether the student's theoretical knowledge or self-reported skills are sufficient or not. For example, the essential ratio can be set to 100%, which would mean the students need to know the answer to all the questions or self-report a defined level of skills for a particular phase. In particular, pre-training assessment should mostly include knowledge quizzes, as students' self-assessment can be misleading [22], [23].

The training phases contain various difficulties, but all on the same topic. The decision component assigns exactly one task from the given phase. This assignment is based on the performance in previous phases and on the pre-training assessment. The performance is measured with time characteristics, used commands, submitted answers, and a solution taken in the phase. The tasks are denoted as T1, T2, ..., TN, where T1 represents the most difficult task in the phase and TN the easiest. Further, the decision component processes the students' performance and knowledge to assign a suitable task from the training phase.

Finally, the post-training questionnaire (Q) is an optional part of training, which enables instructors to collect immediate feedback from the participants.

B. Model

The decision component (P_D) is powered by a mathematical model, which assigns each student the most suitable task in each phase. The model uses binary vectors containing the performance metrics and a list of pre-configured weight matrices to set up the model. We use some of the performance metrics presented in the review of technical metrics for cybersecurity training [24].

Model Formulation: Let us denote the following variables: p , k , a , t , and s are the binary vectors on the correctness or incorrectness of prerequisites for a particular training phase. Vector p is defined as follows: $p = (p_1 \ p_2 \ \dots \ p_m)$, where m is the number of rows. The other vectors use the analogous notation.

- p represents the answers from the pre-training assessment,
- k indicates if the student used the expected key commands in the command line in the given task,
- a denotes whether the student used expected answers to the task,
- t contains the information if the task was completed in a predefined time, and
- s contains the information whether the student asked to reveal the solution for the task,
- W is the matrix with weights for the individual phases' metrics.

The model is defined by the Equations (1) to (3).

By Equation (1), we get the *weight matrix* that is specific for each training phase. The number of weight matrices is

equal to the number of training phases. The weights represent the relationships between phases and their metrics. The value of the weight determines the importance of the metric to the phase. For instance, consider a training with six phases where the third phase deepens the topic exercised in the first phase. In this case, we set the weights in the third matrix so that the selected weights for the metrics from the first phase are non-zero. The other performance metrics with weights set to zero are ignored. The weights have to be manually set by the instructor since each training is unique. The symbols $\alpha, \beta, \gamma, \delta, \varepsilon$ denote the columns in the weight matrices and the $i = 1, \dots, m$ are the rows in the weight matrices.

By Equation (2) we get the *student's performance* based on the defined metrics and their weights for completed phases. The value of the performance is in the interval of $[0, 1]$. In Equation (2), s is multiplied by a , k , and t to distinguish between students who satisfy a , k , and t metrics without using a solution and solved the task on their own.

By Equation (3) we get the *number of the most suitable task* in phase x for a particular student (1 is T1, 2 is T2, and so on).

$$W^{(x)} = \left(w_{ij}^{(x)} \right), i = 1, \dots, m, \quad j = \alpha, \beta, \gamma, \delta, \varepsilon \quad (1)$$

$$f(x) = \frac{\sum_{i=1}^x \left[p_i w_{i\alpha}^{(x)} + s_i \left(k_i w_{i\beta}^{(x)} + a_i w_{i\gamma}^{(x)} + t_i w_{i\delta}^{(x)} + w_{i\varepsilon}^{(x)} \right) \right]}{\sum_{i=1}^x \left(w_{i\alpha}^{(x)} + w_{i\beta}^{(x)} + w_{i\gamma}^{(x)} + w_{i\delta}^{(x)} + w_{i\varepsilon}^{(x)} \right)} \quad (2)$$

$$T_x = \begin{cases} n_x, & \text{if } f(x) \text{ is equal to } 0 \\ \text{trunc}(n_x[1 - f(x)]) + 1, & \text{otherwise} \end{cases} \quad (3)$$

where:

x = the phase a student is entering,

T_x = the most suitable task of the phase x for the student,

n_x = the number of variant tasks in the phase x ,

$$p_i = \begin{cases} 1, & \text{if question group } i \text{ from A is correctly answered} \\ 0, & \text{otherwise,} \end{cases}$$

k_i = commands corresponding to the phase i were used,

e_i = expected time to complete of the phase i ,

o_i = student's completion time in the phase i ,

$$t_i = \begin{cases} 1, & \text{if } o_i < e_i \text{ in phase } i \\ 0, & \text{otherwise,} \end{cases}$$

$$s_i = \begin{cases} 1, & \text{if the solution of the phase } i \text{ is not displayed} \\ 0, & \text{otherwise,} \end{cases}$$

a_i = answers corresponding to the phase i were submitted.

Model Assumptions: The proposed model requires several assumptions that must be met by any system that would use it for hands-on cybersecurity training.

- The learning environment has to collect the required data: commands typed by the students k , phase completion time t , the action of displaying the solution s , the submitted answers a , and the pre-training assessment answers p .
- The model expects that some tasks are related; otherwise, it will heavily rely only on the pre-training assessment that may not be sufficient to capture student's proficiency.
- The pre-training assessment question groups have to be mapped to the training phases to distinguish the level of knowledge and self-reported skills for a particular phase.
- The model assumes that the tasks in the phases are sorted so that the T1 is the most difficult task, T2, ..., TN-1 are easier tasks than T1, and TN is the easiest task.

To ease the unified design and run of the training, we add the following constraints that simplify the model assumptions:

- The students' performance in a phase is evaluated in the same way in all tasks.
- The observed metrics are binary. Other metrics of students' performance, such as similarity of the submitted answers to the correct ones, are either unavailable or ignored.

The model was developed with the aim to reinforce the cybersecurity training with respect to the commonly used performance metrics [24]. Nevertheless, it can be applied in any domain collecting such data.

V. CASE STUDY SETUP

We describe the methods of the case study that uses the proposed adaptive training format and model. The case study uses data collected from 24 participants. The goal is to evaluate whether the proposed format and model are useful for adaptive hands-on cybersecurity training. In particular, we investigate whether the participants' experience is improved and if they successfully finish the training in a timely manner.

A. Teaching Context and Participants

The case study involved three training sessions held remotely in December 2020 and January 2021 at KYPO CRP [5]. 21 participants were undergraduate students of the Masaryk University, and three were graduates with one, two, and 12 years of professional experience in IT. All the participants provided informed consent to use the collected data for research purposes.

We designed a new adaptive training consisting of five interrelated phases. Each phase consists of tasks of various difficulty on the same topic. The phases and tasks were designed by one author and validated by the others. Then, the training was deployed to the KYPO CRP. At the time of the experiment, the learning environment did not provide the support for the proposed adaptive training format (presented in Section IV). We implemented complementary software to process the data required by the model. The data were automatically collected and provided by the learning environment and manually entered into the complementary software by the authors after each phase.

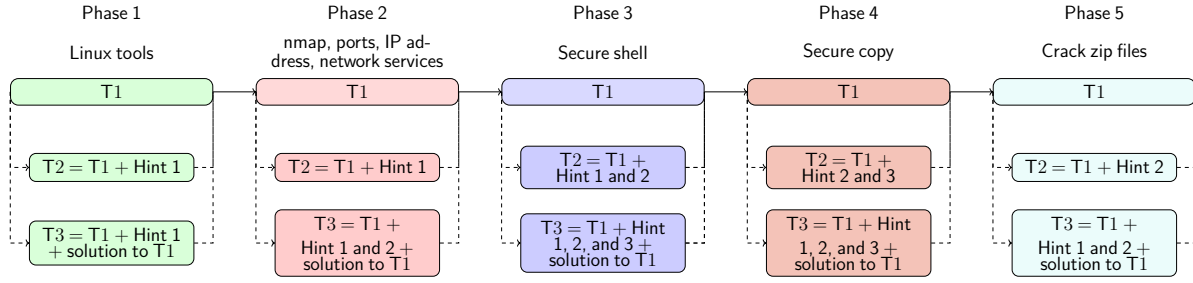


Fig. 3. Phases of the adaptive training instance that follows the proposed generic format. Assignments of tasks contain assignments of base tasks and new or existing hints featured in base tasks.

At the beginning of the training session, students were asked to fill in the pre-training assessment and read the introduction of the training, including all necessary technical settings. Then, we assigned each student the most suitable task from the first training phase computed by the model. Once the student finished the training phase, they notified us, and we asked them to be patient while we entered the data into the complementary software. It calculated the suitable task in the next training phase (this corresponds to the P_D nodes in Figure 2). Finally, after finishing all the training phases, we asked the students to fill in the post-training questionnaire. After the training, all the data were anonymized so that they could not be attributed to a specific participant.

B. Pre-training Assessment

Given the limited time allocated to our training (one and half hours), we used a short pre-training self-assessment presented in Table III. The self-assessment included the following question: *What is your level of skill in the areas below?* and eight areas. The answer *High* means you are able to complete the task very quickly and without much effort. *Medium* means you are able to do it with standard effort. *Low* means you have little experience with that. *None* means you have no experience with that. We considered the students to have sufficient skills if they answered *High* or *Medium*.

TABLE III
WORDING OF PRE-TRAINING ASSESSMENT QUESTIONS AND ANSWERS COLLECTED FROM 24 STUDENTS. ✓ INDICATES A STUDENT SUFFICIENT SKILL AND ✗ INSUFFICIENT.

No.	Question	Phase	Answers	
	What is your level of skill in the areas below:			
Q1	msfconsole interface	none	2 ✓	22 ✗
Q2	WinSCP	none	5 ✓	19 ✗
Q3	build Java projects using Maven	none	12 ✓	12 ✗
Q4	zip and unzip files in CLI	5	15 ✓	9 ✗
Q5	download and transfer files into the server	4	14 ✓	10 ✗
Q6	connect to a server securely	3	16 ✓	8 ✗
Q7	search open ports	2	13 ✓	11 ✗
Q8	basic Linux commands	1	21 ✓	3 ✗

Questions Q4–Q8 were related to topics featured in our training. To avoid the disclosure of the phase topics by the wording of questions in the questionnaire, we added three distractor questions (Q1, Q2, and Q3) about topics not included in the training. The order of the questions differed from the order of the related training phases.

C. Adaptive Training Phases

The training in this study consists of five phases depicted in Figure 3. Each training phase features one base task and two variant tasks. Further, each phase features a task presenting the step-by-step solution. This was a last-resort task for students who would not match any phase prerequisites. In the first training phase, basic Linux tools are practiced in three variant tasks (T1, T2, and T3). Task T2 contains the same assignment as T1 and provides Hint 1. The third task T3 contains the assignment from T1 with Hint 1 and the solution to that task. The subsequent training phases apply the same pattern that differs only in the content of the tasks, hints, and solution provided. The tasks were assigned to each student by the proposed model. The settings of P_D for each training phase are designed using the presented model settings.

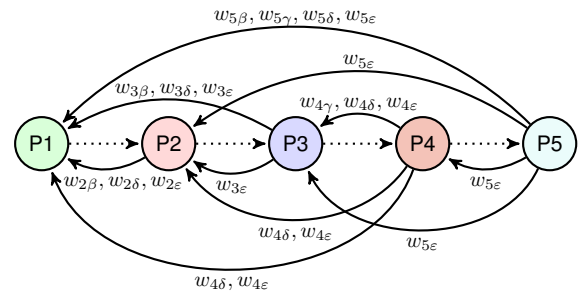


Fig. 4. The relationships between all training phases. P_X is a phase x and w_{xy} is weight for phase x and metric y .

D. Model Settings

To use the model, we must set the weights in the weight matrix \mathbf{W} for each training phase, see Equation (1). These weights indicate the relationships between training phases. For simplicity, we set these weights to zero or one in our case study. One indicates the relationship and zero indicates

that there is no relationship between the phases. Each training phase is related to a particular question group from the pre-training assessment. The relationships between training phases in our training are shown in Figure 4.

VI. RESULTS AND DISCUSSION

In this section, we report the results of the study and summarize our experience with adaptive learning in cybersecurity hands-on training.

A. Adaptive Training Results

Using the ITS terminology, our case study examined *student model* (the participants' performance), *domain model* (the developed training and its phases with tasks), and the *tutor model* (newly proposed model for assigning the most suitable tasks to each participant). Figure 5 shows the transitions of 24 participants between tasks (PXTY) in all training phases. We see that the participants went through different tasks in the training phases, which suggests that the participants' proficiency did not always match the base tasks. We believe this is natural, and the main reason why some participants failed to successfully complete the training in our previous hands-on training sessions.

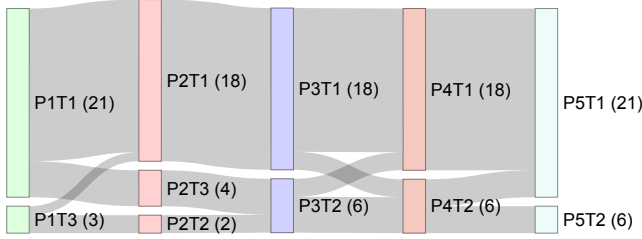


Fig. 5. Transitions of 24 participants between particular tasks in training. PXTY denotes task TY in the phase PX. The number of participants solving the task is in brackets.

The selection of tasks in the first training phase was based on answers from the pre-training assessment because no other performance metrics had been available yet. The three participants claimed that they were not familiar with the Linux operating system, so they played the easiest task in the first phase (P1T3). In the second phase, not only the answers from the pre-training assessment but also the participants' performance from the previous phase were available. The diversity of assignments of tasks to participants increased; the easier tasks were solved by six participants in total. The six participants did not complete the first phase in the expected time (e_i), two used too many commands (k_i), one displayed the solution (s_i), and 11 did not have experience with the tool required for phase two (p_i). It is evident that the participants face different issues during and after the first phase. That confirms our assumption that it is difficult to design static hands-on training suitable for all participants.

In the rest of the training phases, the model assigned the variant tasks to some participants because they were unable to complete the previous phases on time, exceeded the number

of expected key commands (set to 10), or scored low in the pre-training assessment. Overall, even in this relatively small sample of participants, their paths through the training differ substantially. The worst performing participant received mostly the easiest tasks (P1T3, P2T2, P3T2, P4T2, and P5T2) and finished the training in 89 minutes, while the best performing participant completed the most difficult (base) tasks in 13 minutes. Regarding the successful completion of the training, 88% of participants successfully completed the training without any solution taken.

B. Post-training Questionnaire

Immediately after the training session, we asked the participants for their feedback in the online survey. Table IV lists the questions (Q1–Q6) and Figure 6 summarizes the answers.

TABLE IV
WORDING OF THE POST-TRAINING QUESTIONNAIRE.

No.	Question
Q1	Did you feel the tasks were designed so that you can complete the training in a timely manner?
Q2	Did you feel you got stuck at some point during the training?
Q3	How much did you enjoy the training?
Q4	Did you feel the training should be more difficult for you?
Q5	Did you feel you would like the training to be longer with additional tasks to solve?
Q6	Would you like to play more cybersecurity training sessions like this one?

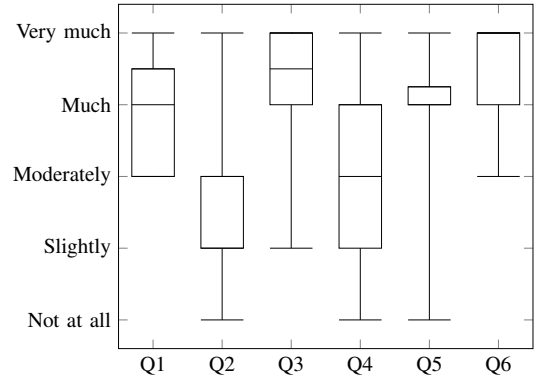


Fig. 6. Answers gathered in post-training questionnaire (n = 24).

In the first question (Q1), the participants reported that the tasks were appropriately designed so that they have successfully completed the training in time. This question has an additional option *Did not receive any*, which was chosen by nine participants. The second question (Q2) was crucial. Five participants reported *Not at all*, eight participants *Slightly*, seven participants *Moderately*, three participants *Much*, and only one participant reported *Very much*. This suggests the training session went relatively smoothly and the majority of participants did not experience anything that would lead them to frustration or premature training termination. In the fourth

question (Q4), only one participant reported that the training should be more difficult. This suggests the need for designing more difficult tasks for the very experienced participants who may get bored if the base tasks are too easy for them. Answers to Q5 indicate that the participants engaged in the training and would like to continue if it would be possible. To conclude, the participants reported (Q3 and Q6) that they enjoyed the training and that they would like to join another adaptive hands-on training in cybersecurity.

C. Limitations

In contrast to other fields, cybersecurity hands-on trainings are usually held in a group of lower tens of participants. Therefore, we believe 24 is a sufficient number of participants to evaluate the created adaptive training format using the newly developed model.

Given the limited time allocated to our training (one and half hours), we used a short pre-training self-assessment. Nevertheless, for training sessions with a larger time allocation, we recommend adding questionnaire quizzes along with self-reported skills [22], [23].

Although the model is not limited to a specific design of variant tasks, we created the variant tasks by changing the text of the assignment (by uncovering particular steps or providing hints). Another option would be to modify the environment (i.e., network and hosts) for the variant tasks. That would give us more freedom in creating the variant tasks.

The model allows including an arbitrary number of tasks in each phase. In our study, we designed three tasks for each phase. Providing more tasks may increase the probability that the participant will get a more suitable task. However, designing more tasks increases instructors' effort to prepare the training.

VII. CONCLUSIONS

Hands-on cybersecurity training sessions usually use static scenarios with limited or no adaptiveness. In this paper, we analyzed student performance and failures in the past training sessions. This led us to propose a new adaptive training format using a graph structure and a generic tutor model. The tutor model is used to assign the most suitable task to each student in each training phase.

Using this innovation, we try to assign the students the optimal path through the training so that they learn as much as possible and keep being motivated for further learning. For these purposes, we developed a new adaptive training format and held three training sessions with 24 participants in total. The results showed that adaptive learning can increase the students' ability to successfully complete the hands-on training, and thus increase the positive students' experience. Further, it showed that the proposed tutor model is useful and can be used for various training sessions with different topics. The students mostly reported that they did not get stuck in any phase of the training and that they enjoyed the training.

To ease the adoption of the proposed innovation, we publish data from the training sessions, together with the model, at [25].

Finally, we provide recommendations for instructors developing adaptive training and ideas for future work.

A. Recommendations for Instructors

To effectively run the adaptive training using the proposed training format and model, consider the following recommendations.

a) *The pre-training assessment questionnaire should be simple and brief:* Cybersecurity education sessions are usually held in a limited time frame. The questionnaire should not consume a large amount of that time, but must still follow best practices for educational assessment [26], [27]. For example, explain the importance of the questionnaire clearly and explicitly to students.

b) *Adjust the weights in the model carefully:* Setting weights in the weight matrices determines the relationships between individual phases and their metrics. Based on that, participant performance for the given phase is calculated. If weights are adjusted incorrectly, the student can get an inappropriate task and may get bored or stuck in the phase.

c) *Design at least three tasks for each phase:* Without enough tasks, the model cannot assign a suitable task for differently performing participants. The base task should be as difficult as possible to target the most experienced participants, and one of the variant tasks should be as easy as possible (step-by-step solution) to encourage less experienced participants.

d) *Allocate more time for students to complete the base phases than you expect:* Since assignments of the base tasks are intentionally vague to allow exploring the phase topic, students need enough time for some trial and error. However, our experience shows that the majority of instructors estimate too short time to complete.

B. Future Work

We proposed a generic model and set its parameters for a particular training session. Therefore, future work should investigate more model metrics and advanced parameter settings. The model decides to move up or down in difficulty for students. But, for example, a student knowing a topic may need a refresher; or a student not knowing a topic may need the challenge to awaken their interest. Further, in our case study, we designed three tasks for each phase and we did not study the effect of a different number of tasks. These issues will be addressed in our future work.

Finally, the decision component (P_D) was provided by the complementary software that required us to do some analytical tasks manually. In our future work, we will enhance this component to be fully automated and integrate it with the KYPO CRP to fully support the proposed adaptive training format.

ACKNOWLEDGMENT

This research was supported by the Security Research Programme of the Czech Republic 2015–2022 (BV III/1–VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20202022158 – Research of New Technologies to Increase the Capabilities of Cybersecurity Experts.

REFERENCES

- [1] D. Mouheb, S. Abbas, and M. Merabti, *Cybersecurity Curriculum Design: A Survey*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 93–107. [Online]. Available: https://doi.org/10.1007/978-3-662-59351-6_9
- [2] M. Bashir, C. Wee, N. Memon, and B. Guo, “Profiling cybersecurity competition participants: Self-efficacy, decision-making and interests predict effectiveness of competitions as a recruitment tool,” *Computers & Security*, vol. 65, pp. 153–165, 2017.
- [3] R. S. Putri, A. Purwanto, R. Pramono, M. Asbari, L. M. Wijayanti, and C. C. Hyun, “Impact of the COVID-19 pandemic on online home learning: An explorative study of primary schools in Indonesia,” *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 4809–4818, 2020.
- [4] C. Braghin, S. Cimato, E. Damiani, F. Frati, L. Mauri, and E. Riccobene, “A Model Driven Approach for Cyber Security Scenarios Deployment,” in *Computer Security*, A. P. Fournaris, M. Athanatos, K. Lampropoulos, S. Ioannidis, G. Hatzivasilis, E. Damiani, H. Abie, S. Ranise, L. Verderame, A. Siena, and J. Garcia-Alfaro, Eds. Cham: Springer International Publishing, 2020, pp. 107–122.
- [5] J. Vykopal, P. Čeleda, P. Seda, V. Švábenský, and D. Továřík, “Scalable Learning Environments for Teaching Cybersecurity Hands-on [in press],” in *Proceedings of the 51st IEEE Frontiers in Education Conference*, ser. FIE ’21. New York, NY, USA: IEEE, 10 2021, pp. 1–9.
- [6] K. Colchester, H. Hagra, D. Alghazzawi, and G. Aldabbagh, “A Survey of Artificial Intelligence Techniques Employed for Adaptive Educational Systems Within E-Learning Platforms,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 1, pp. 47–64, 2017.
- [7] J. R. Carbonell, “AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction,” *IEEE transactions on man-machine systems*, vol. 11, no. 4, pp. 190–202, 1970.
- [8] National Academy of Engineering. (2008) NAE Grand Challenges For Engineering. National Academy of Engineering. [Online]. Available: <http://www.engineeringchallenges.org/9127.aspx>
- [9] R. A. Sottilare, A. C. Graesser, X. Hu, A. Olney, B. Nye, and A. M. Sinatra, *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling*. Adelphi, Maryland: US Army Research Laboratory, 2016, vol. 4.
- [10] Y. Long and V. Aleven, “Students’ Understanding of Their Student Model,” in *Artificial Intelligence in Education*, G. Biswas, S. Bull, J. Kay, and A. Mitrovic, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 179–186.
- [11] Q. Hu, Y. Huang, and Y. Li, “A “Content-Behavior” Learner Model for Adaptive Learning System,” in *Intelligent Computing Methodologies*, D.-S. Huang, K.-H. Jo, and L. Wang, Eds. Cham: Springer International Publishing, 2014, pp. 466–474.
- [12] R. Nkambou, R. Mizoguchi, and J. Bourdeau, *Advances in Intelligent Tutoring Systems*. Berlin/Heidelberg, Germany: Springer Science & Business Media, 2010, vol. 308.
- [13] T. Effenberger and R. Pelánek, “Measuring Students’ Performance on Programming Tasks,” in *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, ser. L@S ’19. New York, NY, USA: Association for Computing Machinery, 2019.
- [14] H. Khosravi, S. Sadiq, and D. Gasevic, “Development and Adoption of an Adaptive Learning System: Reflections and Lessons Learned,” in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 58–64.
- [15] G. Hatzivasilis, S. Ioannidis, M. Smyrlis, G. Spanoudakis, F. Frati, L. Goeke, T. Hildebrandt, G. Tsakirakis, F. Oikonomou, G. Leftheriotis et al., “Modern Aspects of Cyber-Security Training and Continuous Adaptation of Programmes to Trainees,” *Applied Sciences*, vol. 10, no. 16, p. 5702, 2020.
- [16] J. Paladines and J. Ramirez, “A Systematic Literature Review of Intelligent Tutoring Systems With Dialogue in Natural Language,” *IEEE Access*, vol. 8, pp. 164 246–164 267, 2020.
- [17] A. Alkhatlan and J. Kalita, “Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments,” *International Journal of Computer Applications*, vol. 181, no. 43, pp. 1–20, Mar 2019. [Online]. Available: <http://www.ijcaonline.org/archives/volume181/number43/30402-2019918451>
- [18] S. Nebel, M. Beege, S. Schneider, and G. D. Rey, “Competitive Agents and Adaptive Difficulty Within Educational Video Games,” *Frontiers in Education*, vol. 5, p. 129, 2020. [Online]. Available: <https://www.frontiersin.org/article/10.3389/educ.2020.00129>
- [19] S. Xue, M. Wu, J. Kolen, N. Aghdaie, and K. A. Zaman, “Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW ’17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 465–471. [Online]. Available: <https://doi.org/10.1145/3041021.3054170>
- [20] Circadence. (2020) Circadence: Cyber Learning Platforms. Circadence. [Online]. Available: <https://www.circadence.com/>
- [21] 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: USENIX Association, Aug. 2016. [Online]. Available: <https://www.usenix.org/conference/ase16/workshop-program/presentation/vykopal>
- [22] V. Švábenský and J. Vykopal, “Challenges Arising from Prerequisite Testing in Cybersecurity Games,” in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 56–61. [Online]. Available: <https://doi.org/10.1145/3159450.3159454>
- [23] J. Mirkovic and P. A. Peterson, “Class capture-the-flag exercises,” in *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.
- [24] K. Maennel, “Learning Analytics Perspective: Evidencing Learning from Digital Datasets in Cybersecurity Exercises,” in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2020, pp. 27–36.
- [25] P. Seda, J. Vykopal, V. Švábenský, and P. Čeleda, “Dataset: Reinforcing Cybersecurity Hands-on Training With Adaptive Learning,” 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.5011504>
- [26] A. W. Astin and A. L. Antonio, *Assessment for Excellence: The Philosophy and Practice of Assessment and Evaluation in Higher Education*. Lanham, Maryland: Rowman & Littlefield Publishers, 2012.
- [27] G. Petty, *Teaching Today: A Practical Guide*. Cheltenham, Great Britain: Nelson Thornes, 2009.