Bridging the Gap: Adapting a Security Education Platform to a New Audience

Gian Marco Mennecozzi
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
gmm@es.aau.dk

Kaspar Hageman
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
kh@es.aau.dk

Thomas Kobber Panum
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
tkp@es.aau.dk

Ahmet Türkmen
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
atu@es.aau.dk

Rasmi-Vlad Mahmoud
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
rvm@es.aau.dk

Jens Myrup Pedersen
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
jens@es.aau.dk

***Abstract*—The current supply of a highly specialized cyber security professionals cannot meet the demands for societies seeking digitization. To close the skill gap, there is a need for introducing students in higher education to cyber security, and to combine theoretical knowledge with practical skills. This paper presents how the cyber security training platform Haaukins, initially developed to increase interest and knowledge of cyber security among high school students, was further developed to support the need for training in higher education. Based on the differences between the existing and new target audiences, a set of design principles were derived which shaped the technical adjustments required to provide a suitable platform - mainly related to dynamic tooling, centralized access to exercises, and scalability of the platform to support courses running over longer periods of time. The implementation of these adjustments has led to a series of teaching sessions in various institutions of higher education, demonstrating the viability for Haaukins for the new target audience.**

***Keywords*—cyber security teaching, capture the flag, active learning**

I. Introduction

The demand for cyber security professionals has been increasing during the past years and is expected to increase even more in the future [1]. To meet this increased demand, various higher education institutions globally have started to develop curricula specifically to educate engineers in cyber security, while others have integrated cyber security into their existing teaching curricula [2], [3]. Practical exercises are an essential part in such curricula, and therefore several educational teaching platforms have been designed over the past years [4]–[7] to help out the learning process of students and encourage them to pursue a career in information security.

These platforms provide an environment that can be used for teaching and training purposes in cyber security courses by simulating real vulnerable systems and supporting complex cyber scenarios, as well as training users in monitoring and defending cyber infrastructure against malicious activities. This allows students to execute attacks on these systems without harming actual systems, and to obtain both an offensive and defensive perspective of security practices.

Most of these platforms are designed to support a common exercise format, Jeopardy Capture-The-Flag (CTF) [8] which nowadays is a well known format to demonstrate the user skills in solving cyber security tasks and problems. In this format, the participants are tasked to find *flags* (i.e. a hidden string of text) by successfully exploiting vulnerable computer systems, being awarded points for each flag they discover. CTF competition brings several advantages when applied in teaching environments [6]. For example, these types of competitions allow students to legally hack systems in a safe environment, by identifying vulnerabilities and trying to compromise them, while also allowing them to learn to defend against attacks [9]. CTFs have also been shown to be effective in keeping the students engaged by their hands-on nature and through the entertaining experience [10]. When involved in a CTF, being able to work as a group in a team is important for students to achieve their goals, leading the students to improve their communication skills, and to share, compare and broaden their knowledge [11]. Moreover, challenge based learning stimulates the development of problem solving skills leading the students to be involved in finding better solutions [12].

Although these existing educational teaching platforms contribute to the learning process, none of them automate the process of creating custom vulnerable environments for teaching classes for high schools students. This led Aalborg University, Denmark, to develop the first version of Haaukins [13], an educational tool used for conducting short training events at high schools. These events, usually running between two hours up to a few days, were intended to engage students without prior information security knowledge, in a new topic, and to generate interest in a future career into the security field.

Since its launch, Haaukins has proven to be successful in high schools and attracted attention from other Danish institutions within higher education. In order to increase the

usage of the platform, Aalborg University together with other educational institutions has started to adapt Haaukins from both a technical and an education perspective to accommodate IT and engineering students within higher education. The main contributions of this paper are as follows:

- Define a formalization of the differences in teaching environment, across high school and higher levels of education, for cyber security education.
- Present a set of design goals, based on the formalization, that the Haaukins platform is required to adopt to address these differences.
- Develop a solution for these design goals, that have been integrated into the existing open source platform.

The rest of this paper is organized as follows. A background of the initial design of Haaukins is presented in Section II, followed by a comparison with the new target audience in Section III. In order to adapt the platform to the new target audience, a list of design principles is presented in Section IV, which is followed by the fulfilment of those principles in Section V. The platform deployment is presented in Section VI, followed by the conclusion in Section VII.

II. Background

Besides existing educational platforms there are a variety of commercial platforms where it is possible to practice cyber security in a CTF format, including 'Hack the Box' [14] and the more recently developed 'Try Hack Me' [15]. Whereas 'Hack the Box' provides challenging scenarios fitting for a more experienced target audience, 'Try Hack Me' provides several learning paths suitable for introducing beginners to the basics of security. However, such commercial platforms generally have no option to expand upon the training material and scenarios provided (such as adding more vulnerable machines), which is a severe restriction when teaching courses according to specific learning goals and objectives. An educational institution must have the opportunity to tailor the teaching material to their curriculum, and as such cannot rely on closed platforms.

'PicoCTF' [16] is a platform developed by Carnegie Mellon University that achieved success during the past years. In order to encourage cyber security interest among high school students, it provided an interactive game and a terminal user interface used to interact with the exercises. Similar to Haaukins, it is open source and has it has been created for high schools students. However, in contrast to Haaukins, it constraints itself to exercise types which students can do on any computer systems, and without relying on professional tooling.

Setting up an environment composed of computer systems, vulnerable hosts and connections between them is time-consuming and errors can be hard to handle. Haaukins aims to facilitate the learning process by helping teachers to automate the tedious setup and management of those environments, by making them accessible with no prior, complex configuration. This allows students to have their own virtual and isolated environment to practice cyber security skills, while having the convenience of accessing it from their own devices simply through any web browser. High school students who, due to their limited amount of experience with cyber security and a limited understanding of underlying computer science topics in general, need an automated and highly accessible way to access those environments. Moreover, in order to support other educational institutions in conducting short CTF workshops, Haaukins has been made available as an open-source project on GitHub[1] with a GNU GPLv3 license.

Haaukins is implemented as a Jeopardy-style CTF platform, in which the teacher can create an event and students access their own individual environments, referred to as *lab*s, created within the scope of that event. Each virtual *lab*, accessible through a student's laptop, consists of a computer network that has multiple computer systems, which are under control of the student or student group. Within this *lab*, students have to work towards solving a set of *exercises*, where each of them are related to a specific concept within information security. Most of the exercises are designed with specific learning objectives in mind. Exercises are solved by exploiting vulnerabilities of computer systems in the virtual labs; vulnerabilities that are intentionally embedded in the computer systems by the exercise developer. In order to exploit these vulnerabilities, the students must understand the underlying problem with the software, which makes these exercises a valuable teaching tool. Each event can be composed of any set of exercises that the teacher wishes to use in that session. Figure 1 illustrates the interaction between teacher, students and Haaukins itself. A website, which is automatically generated per event, provides both teacher and students with information about the event and its exercises.

The high school target audience required four main design goals to be fulfilled, that shaped the development of Haaukins in the very early stage of its existence. Those design goals are described in [13] and are summarized as follow:

a) Fully Automated: Haaukins was designed to automate the lab configuration process completely, and to do so in a relatively short time, making the preparation of an event painless.

The automation process will start and connect the required instances and components in order to have an environment available for the learning aspects. This process eliminates the need for manual configuration and provide error handling of labs for the teachers.

b) Transparent: As students are more inclined to share solutions than in a competition setup, potentially leading to cheating that negatively influences their learning experience, a unique way of creating labs has been provided in Haaukins. For each lab, the flags to be found are unique, reducing the possibility of sharing flags. In addition, the platform monitors all actions that students make within it, allowing for the analysis of this afterwards.

c) Highly Accessible: Given the short nature of the training events at high schools, it was considered imperative

[1] https://github.com/aau-network-security/haaukins

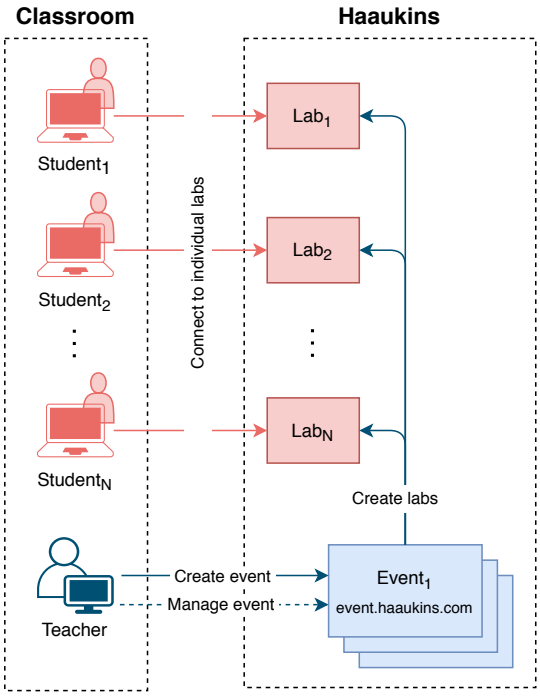

Fig. 1. The interaction between a teacher, a classroom full of students and the Haaukins components in the first version of the platform

that students spend as little effort as possible on establishing access to their labs. In this case, Haaukins allows participants to access their virtual lab (1) quickly (i.e. in a matter of minutes), (2) without prior knowledge of accessing a virtual platform and (3) independently of their physical location and the operating system they use (i.e. access from anywhere via their Internet browser).

d) Realistic: In order to teach students skills that are usable outside of the classroom, the platform was required to reflect a real world setting. The existing pool of exercises was thus designed to replicate real-life situations, and the platform had to support these types of exercises. In Haaukins all the exercises created are from real-life situations and interaction's realism is kept in order to ensure that students are gaining useful skills. Additionally, the taught techniques and available tool kit are identical to those used by security professionals.

III. Target Audience

During the last year, the usage of Haaukins increased because of interest from new educational institutions. From the feedback received after every event, we have realised that the platform was able to satisfy the design goals allowing teachers to successfully conduct cyber security courses. The increase in usage has brought more visibility to the platform, thus leading to new expectations and new plans for its continuous development. Haaukins, could potentially be used in higher education as well, however due to the different target audience, it would need several improvements in order to make the new users comfortable with the platform.

This section seeks to clarify details of the new target audience, and their distinct traits, when compared to the traditional target audience for Haaukins. As previously stated, the audience intending to use the adjusted platform is *students in higher education*. To further clarify the expected skill set of this group, we define it as students who have attended at least one semester in computer science, computer engineering or comparable educations at least at undergraduate level. Concretely, students within this group should have a novice level of understanding of fundamental computer science topics, such as: (1) networking, (2) operating systems, and (3) programming. This thereby puts this group of students, in terms of competences, ahead of the original target audience of Haaukins (i.e. high school students).

Consequently, the established experience of the new target audience is reflected in their ability to interact with and understand computer systems.

Trait 1. *Higher education students are expected to have preliminary knowledge of various tools, selected to their preference, to diagnose and interact with computer systems and computer networks.*

This is an important trait that differs from the original high school setting, where students are expected to have no or very limited technical understanding. This additional knowledge is also reflected in the type of teaching material to be used for the new target audience.

Trait 2. *With a deeper technical knowledge, students of higher education are able to work on more complex exercises that involve understanding and attacking highly-interconnected and complex multi-computer systems.*

From a teacher perspective, designing such exercises can be challenging and time consuming, as ensuring the interconnection of a multitude of computers correctly is a complex task. These exercises typically encapsulate some intended-by-design vulnerabilities, to be found by the students, and the discovery of the given vulnerabilities serve as an objective of the exercise. Students are typically expected to interact with the computers of the systems, of the exercise, in an offensive and destructive manner. Therefore exercise designers must avoid unexpected vulnerabilities in the systems to be attacked, since such unexpected vulnerabilities can potentially halt the completion of the intended exercise. From a course point of view, the duration itself is often longer than the high schools events, and a class might run just once per week through a semester. As a consequence of this, reliability and availability must be provided even for such long term events, so the users of the platform do not experience interruptions and resets of events, users or exercise progress.

Trait 3. *Given the multitude of attack vectors that a computer system can have, with the typical length and size of a university course, keeping each students lab in a healthy state (ability to complete exercises) is challenging.*

The identified and distinct traits of the new target audience

captures the challenges sought out to be addressed by technical solutions that can be integrated into the existing Haaukins platform, thus expanding its application domain. It is important that the application domain is expanded rather than changed, and that Haaukins still fulfill the requirements for the original target audience. The improved platform thus also appeals to the groups in-between, i.e. students in higher education without the competences listed above.

IV. Design Principles

The three traits identified in Section III, serve as the problems which have to be addressed in order for Haaukins to viable for the new teaching context. Prior to the technical design, we seek to put forward three design principles that will drive the changes to the existing platform. These design principles have to coexist with existing ones (covered in Section II), thus their compatibility to the existing ones are discussed.

Haaukins provides rapid access to students' labs remotely (through a web browser) to a computer with pre-installed tools, serving as a fixed toolbox, used for accessing the lab environment and interacting with the exercises. This design was implemented in relation to the existing design principle of being highly accessible. However, the design of having a fixed toolbox (static tooling) could potentially violate Trait 1, as the desired set of tools might not match the provided set of tools. In order to address this problem, and in contrast to the existing static tooling currently provided, we present the following design principle:

Design Principle 1. [Dynamic Tooling] Students should be able to use their desired set of tools within their lab of exercises.

Having dynamic tooling could potentially be an undesired feature for the original teaching context (high schools), which includes more novice students that have less preliminary competences in the field. Thereby, it is important to ensure that this principle serves as an alternative, such that it can coexist with the current method of "static tooling". Knowing that the teaching context includes students with more preliminary knowledge (Trait 2), forces the exercises containing computer systems to become larger (groups of computers) and more inter-connected (network communication). These types of exercises are naturally costly to design due to their complexity, and the current architecture for Haaukins relies on teachers, on an individual level, to implementing these. Thereby, as a measure to reduce this cost across, we propose the following design principle that is based on sharing resources:

Design Principle 2. [Centralization of Exercises] Complex exercises should be provided by a centralized source, such that teachers across various institutions can share implementations of exercises, thus enabling future teachers to benefit from existing work.

Hosting and serving these exercises in individual labs for each student in a university class is a vastly different scale when compared to the high school setting (Trait 3). Moreover the platform has to support a higher volume of concurrent students coming from both the previous and new target audience. Scaling the platform to support a teaching context of a university class (more than 100 students) and an increased amount of events running at the same time, is a two-fold challenge. Indeed, both the technical capabilities (efficiency, resources) and orchestration of labs (maintenance, restarts) are required to scale to this new volume of students. Moreover teachers should have the possibility to monitor and manage student labs from a highly user friendly interface while the events are running.

Design Principle 3. [Scalability] The platform should provide both computational efficiency (technical scaling) and orchestration features (teaching scaling) that ensure exercises within labs remain healthy and available while several events are running concurrently.

V. Evolved Haaukins platform

The design principles defined in the previous section aim at improving Haaukins in order to make the new target audience comfortable in using the platform. The evolved platform should address the requirements of students in higher education without impacting the usability for high schools, which are still a core user group of Haaukins. This chapter presents a list of improvements implemented based on each of the principles defined in IV.

Dynamic Tooling. In order to adhere to Design Principle 1, two alternatives were considered; either to integrate more tools in the lab or to provide an alternative access method to the lab. In the first approach, a more customizable lab environment can be provided to the users, e.g. by giving the teachers or students themselves the option to compose a toolkit from a curated list of tools and operating systems. This curated list would have to encompass all possible tools that students would like to use, which in practise would be difficult to accomplish. The second alternative instead relies on giving the students access to the lab through a different method than the browser-based method. Previously, students would remotely control a computer system prepared specifically for this, and these systems were identical across all labs in an event, leaving little room for customization. Instead, students could be given access to a *network connection* to the lab, thereby opening the option for students to connect their own computer systems to the labs, with their own custom tooling. In order to respect all previous and newly defined design principles, the platform must allow the user to still have a fully automated configuration process and a highly accessible way to the exercises.

From a technical point of view, the choice fell on the integration of a Virtual Private Network (VPN) [17], and specifically Wireguard [18], in which a secure connection to another network over the Internet is created, in our case to the *lab*. A VPN connection can be established from any operating

system from any geographic location, merely requiring the students to configure their local Wireguard with a configuration file provided by Haaukins. Similar to the previous web-based access method, a VPN connection can generally be established from computer networks without requiring changes to the IT infrastructure. As a result, this solution does not violate the original highly-accessible design principle, but contributes to Design Principle 1. Although configuring a VPN is considered a fairly simple setup step for an experienced student, this it not the case for the traditional target audience, and therefore Haaukins supports both (1) the web-based access method and (2) the VPN connection, and a teacher can make a per-event decision as to what access method suits the target audience best.

Centralization of Exercises. The first version of the platform has been developed to allow teachers to run events using either custom exercises or readily-available exercises provided by the platform itself through an exercise library. In the first case, teachers could create specific exercises for their courses and use them in their events in order to teach different topics not already provided by the platform. Although this feature brought more flexibility to the platform when referring to high schools, university teachers could not benefit much of it. This is largely due to the fact that different target audiences rely on different exercises that differ from each other in terms of both content and complexity (Trait 2). Exercises for higher education students are not only harder to solve compared to those for high school students, they are also more complex and take longer more time and efforts to create. These exercises, in fact, have to be created either focusing on a specific topic and go into details or have a broad approach where it covers several topics. In both cases, the teacher has to design and create several steps of difficulty in order to let students improve their skills and time spent on the exercise.

Creating an exercise for a course is time-consuming, especially for the more advanced exercises which are to be used within higher education, and therefore is not always an option. To support teachers, many exercises have been created and made available to the use in their events, and a clear workflow has been established for creating, testing and including exercises for those who want to create their own. The main goal was to provide a centralized pool of exercises with different content and of different complexities where teachers can choose according to their courses (Design Principle 2). Each exercise is accompanied by a description, and a list of prerequisites and outcomes has been made in order to facilitate teachers in choosing a relevant composition of exercises for their courses. Finally, to facilitate an even better exercise selection phase, the exercises have been grouped based on different difficulty levels (e.g. the number of steps needed to solve it and the topic covered) and divided into different categories that cover different fields of cyber security (e.g. web exploitation, forensics, binary, reverse engineering and cryptography).

Scalability. Haaukins must also support managing a higher amount of events running at the same time. In order to provide a reliable and fault tolerance platform, it has to scale in two main directions (Design Principle 3) described as follows.

a) ***Teaching Scaling****:* In order to maintain the labs healthy and available, a "reset functionality" has been created, available to both teachers and students in order to restart (i.e stop and start) *lab*s as well as individual exercises in case of crashes, which can happen if a student make mistakes when attempting to solve an exercise or when exercises are not properly developed with the destructive behaviour of the teaching context in mind (Trait 3). In such cases, one or more exercises in the *lab* are reverted back to the initial state right after the lab was created. The students lose their progressions towards this specific exercise, but it allows them to experiment with potential destructive offensive techniques that will break the exercise. In fact, as exercises largely focus on breaking existing computer systems, those systems are brought to a high degree of stress. As such, platform allows a graceful recovery from errors, instead of burdening teachers with developing bulletproof exercises. This functionality has been made available on the event website for the students while for teachers it has been implemented in the *web client*.

In the previous version of Haaukins, the event creation and management controls were provided via a command-line interface (or *cli*), that had to be downloaded and installed on the computer of the teacher. This command-line program could be used to send some basic commands remotely to the physical server on which Haaukins was running, thereby managing events. Prior to be able to use this program, a teacher had to be granted access by another teacher as a security mechanism, which introduced another barrier for quickly setting up an event. Feedback from high school organizers showed that this approach was not user-friendly enough, and that it was too time-consuming to use.

In order to overcome this issue a different way to interact with the platform had to be provided, and a user interface *web client* connected to the platform was created. In detail, the *web client* is the web-application version of the *cli* which provides the same functionalities of the *cli* along with a number of new functionalities designed to improve scalability as well as the overall organisation experience. With this solution it is no longer needed to download and install the *cli* on the teachers computer machine. Teachers can access the *web client* upon request in order to create and manage their own events no matters where they are - simply by their web browser of choice. From an intuitive user interface, the teachers are able to choose the event configuration (e.g. event name, event capacity, exercises and VPN option) and check the status of the teams signed up in their events.

The *web client* is linked to the centralized exercises pool thus allowing teachers to insert new custom exercises and get all information about already existing and available exercises. This connection aims to facilitate the teacher in choosing the relevant exercises for his or her event.

Besides management of events and their respective *lab*s, Haaukins has the ability to monitor and log student's interac-

tions with the platform, which enables the ability to identify if the participants become stuck while exploiting exercises. This functionality is only activated if consent is granted from individual students, and is implemented by storing the stream of key presses to log files, that serve as the basis of the analysis of behavior.

*b) **Technical Scaling**:* The new target audience will bring with it not just more events running simultaneously on the server, but also larger events due to the higher number of students for each course, thus leading to a higher computation load on the platform. A potential problem that might occur because of this higher demand, is that the platform might not have sufficient capacity to be able to manage all the events thus leading to the rejection of some of them. A main goal is therefore to provide both target audiences with a platform that is able to support all the requested events without affecting the performance of the platform or other events (Design Principle 3).

To meet this goal, two main approaches to make the platform more scalable have been evaluated, and both horizontal and vertical methods have been taken into consideration: the horizontal approach relies on replicating the platform on multiple servers, thus leading to a distributed version where events can run in different servers. The vertical approach instead consists of adding more resources (i.e. memory and hard disk) to the current server in order to make it more powerful. From our point of view both methods where suitable for the platform, the former being more expensive in terms of time due to the refactoring the code base of the platform but cheaper in terms of the monetary cost, while for the latter it is the other way around.

Also considering the possibility to make a platform cloud based, more effort has been made in making the platform available in a distributed way without affecting the usability for teachers and students. In this sense the platform has been split in microservices [19] running on different servers, which also allows for a more easier deployment to the cloud in the future [20]. The vertical approach has been applied as well on the main server, where more resources have been installed.

From previous experience with high school events, it was found that labs in events that last longer than two weeks have a far lower resource utilization (i.e. the percentage of time that a lab is actively being used) than shorter events. In fact, some of the labs were not used for several full days before being used again afterwards, occupying resources of the host server and consequently potentially refraining other students from using the platform. As described in Trait 3, this scenario might occur more often, eventually denying requests for events due to the limited capacity of the server hosting the platform. Although the technical scaling improvements aim to provide the opportunity to everyone to use the platform, those long events might thus cause a problem. To overcome this issue, a 'sleep mode' feature has been developed, which automatically suspends *lab*s that have not been used for a while and resume them when the students log into the event again. This feature aims to save resources on the server - especially for the new usage - and thus boosts the scalability of the platform.

VI. Deployment in Higher Education

Throughout the development of the evolved Haaukins platform, it has been used in various settings within higher education, and feedback has been collected as input to the development process. The usage includes courses within two universities and four university colleges, as well as larger events in the framework of higher education, such as summer schools and conferences. It was also used in university-facilitated events for IT professionals in companies including sectors such as finance, energy, IT and national authorities.

While different events and courses were organised differently, in general three steps were included: (1) In the preparation phase, the course or event was planned and set up. This includes choosing relevant exercises, determining whether VPN or web browser access should be used. In the beginning, this was in most cases done in close collaboration with the Haaukins developers, but as more teachers gained experience in using the platform and as the improvements described in this paper were developed - in particular the *web client* - this was increasingly done by the teachers independently. (2) In the next phase, the event/course was held. Unless any problems arose, this was usually done by the teachers. (3) Finally, in the evaluation phase, feedback was collected from the teachers and/or the students.

The feedback collected included the experience from both phase (1) and (2) together with general feedback and suggestions about the platform. These collection of feedback has served two purposes. One purpose was to use it for input to the overall platform design and development, where the resulting changes would be of a more fundamental character. Such changes would be incorporated to the overall development road map, which was discussed among partner institutions of higher education at regular meetings. Another purpose was to identify issues where smaller adjustments could improve the user experience. In many cases, these were straightforward to implement, e.g. better explanations of platform usage and exercises. By the time this paper was submitted, Haaukins has been used by more than 1.000 students from different target groups.

VII. Conclusion and Future Work

In this work, we presented an evolved version of Haaukins, a cybersecurity training platform that facilitates the learning process by helping teachers in creating cybersecurity training scenarios in a secure, closed and virtualized environment. Over the last years, the platform has been used in several high schools with consistently positive feedback and it was decided to improve the platform in a way that would support the usage in higher education.

The new target audience, compared to the previous one, has been identified as a more experienced student who is able to interact with computer systems and computer networks using various tools, and who is able to address more complex exercises. Due to those differences and the typical length and

size of a university course, a list of design principles, which have to coexist with the existing ones, have been defined and afterwards shaped into technical improvements on the platform.

Examples of such improvements include that a VPN connection has been provided as an alternative way to connect the *lab*s, thus enabling the students to use their own tools. An exercises pool has been made available for teachers in order to let them benefit from already made exercises, thus avoiding the time invested in creating them. Finally, the platform has been made more scalable in order to handle a higher amount of students and longer events running at the same time.

These collective changes made Haaukins a platform for both students of higher education and high schools students, driven by an increased interest by schools in Denmark. The platform is currently being under further development to widen its appeal to even more target audiences and to be used on a larger scale, and additional research studies are also being undertaken: In order to obtain a better understanding of the challenges of the game based learning experience, we are planning to conduct user studies in the near future. Moreover an investigation of which exercises should be developed to ensure a good progression in the learning path of different students will be carried out.

VIII. Acknowledgment

The authors would like to thank The Danish Industry Foundation for the continued support to the development of Haaukins.

References

[1] C. Ventures, "Cybersecurity jobs report," *Herjavec Group*, 2017.

[2] E. C. 2018. (2018) Resilience, deterrence and defence: Building strong cybersecurity in europe. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/resilience-deterrence-and-defence-building-strong-cybersecurity-europe

[3] A. University. (2020) Cyber security, msc in engineering. [Online]. Available: https://www.en.aau.dk/education/master/cyber-security/

[4] P. Chapman, J. Burket, and D. Brumley, "Picoctf: A game-based computer security competition for high school students," in *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[5] E. Trickel, F. Disperati, E. Gustafson, F. Kalantari, M. Mabey, N. Tiwari, Y. Safaei, A. Doupé, and G. Vigna, "Shell we play a game? ctf-as-a-service for security education," in *2017 {USENIX} Workshop on Advances in Security Education ({ASE} 17)*, 2017.

[6] A. Mansurov, "A ctf-based approach in information security education: an extracurricular activity in teaching students at altai state university, russia," *Modern Applied Science*, vol. 10, no. 11, p. 159, 2016.

[7] L. Tobarra, A. P. Trapero, R. Pastor, A. Robles-Gómez, R. Hernández, A. Duque, and J. Cano, "Game-based learning approach to cybersecurity," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 1125–1132.

[8] T. Chothia and C. Novakovic, "An offline capture the flag-style virtual machine and an assessment of its value for cybersecurity education," in *2015 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*, 2015.

[9] S. Wi, J. Choi, and S. K. Cha, "Git-based {CTF}: A simple and effective approach to organizing in-course attack-and-defense security competition," in *2018 {USENIX} Workshop on Advances in Security Education ({ASE} 18)*, 2018.

[10] M. Katsantonis, P. Fouliras, and I. Mavridis, "Conceptual analysis of cyber security education based on live competitions," in *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2017, pp. 771–779.

[11] J. Werther, M. Zhivich, T. Leek, and N. Zeldovich, "Experiences in cyber security education: The mit lincoln laboratory capture-the-flag exercise." in *CSET*, 2011.

[12] R. S. Cheung, J. P. Cohen, H. Z. Lo, and F. Elia, "Challenge based learning in cybersecurity education," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer ..., 2011, p. 1.

[13] T. K. Panum, K. Hageman, J. M. Pedersen, and R. R. Hansen, "Haaukins: A highly accessible and automated virtualization platform for security education," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161. IEEE, 2019, pp. 236–238.

[14] Hack The Box Ltd, "HackTheBox.eu." [Online]. Available: https://www.hackthebox.eu/

[15] Try Hack Me - London, "Try Hack Me." [Online]. Available: https://tryhackme.com/

[16] Carnegie Mellon University, "picoCTF." [Online]. Available: https://picoctf.org/

[17] P. Ferguson and G. Huston, "What is a vpn?" 1998.

[18] J. A. Donenfeld, "Wireguard: next generation kernel network tunnel (2018)," *URL https://www. wireguard. com/papers/wireguard. pdf. version 416d63b*, pp. 06–30, 2018.

[19] N. Alshuqayran, N. Ali, and R. Evans, "A systematic mapping study in microservice architecture," in *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, 2016, pp. 44–51.

[20] V. Singh and S. K. Peddoju, "Container-based microservice architecture for cloud applications," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 847–852.