Interactive Cybersecurity Education for Upper Secondary School

Anonymous Author 1
E-Mail-Address
CS Department
University
Town, Country

Anonymous Author 2
E-Mail-Address
CS Department
University
Town, Country

Anonymous Author 3
E-Mail-Address
CS Department
University
Town, Country

Abstract

We introduce MockHack [name changed], an interactive learning platform for teaching cybersecurity to upper secondary school students. It is fully web-based, not requiring additional software besides a browser and providing as low of an entry barrier as possible for novices. Two main components form the platform; the lesson center and the "target applications." The lesson center holds the theoretical lessons, questions, exercise tasks that require an interaction with the target applications, as well as progress tracking and classroom management for teachers. The target applications are sandbox environments that mimic common real-world apps that are interesting for educational purposes.

In a case study, we instantiate the target application as a small mock social media network, completed with accompanying lessons about web security and social media awareness.

Besides an introduction to the implementation, we report our findings on semi-structured interviews with experts in cybersecurity, as well as on two surveys filled out by teachers and students, respectively, to evaluate their feedback about the platform.

CCS Concepts

• Applied computing → Interactive learning environments; **• Information systems → Social networks**; **• Social and professional topics → K-12 education**; **• Security and privacy**;

Keywords

K–12; interactive learning environments; cybersecurity literacy

ACM Reference Format:
Anonymous Author 1, Anonymous Author 2, and Anonymous Author 3. 2026. Interactive Cybersecurity Education for Upper Secondary School. In *Proceedings of the 31st Annual ACM Conference on Innovation and Technology in Computer Science Education, July 13–15, 2026, Madrid, Spain.* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Bringing effective cybersecurity education into secondary schools is challenging due to various factors, such as the rapid increase in technical complexity of computer systems, or ethical issues when teaching cybersecurity strategies to students. To possibly solve these issues, schools might opt for specialized systems which offer reduced complexity and simplified vulnerabilities for guided novices. Unfortunately, setting up these systems—or even creating them from scratch—is a challenging task in itself and outside the scope of most schools. For this purpose, we introduce a dedicated web-based platform, MockHack [name changed], which provides specialized modular sandbox environments and suitable tools for in-class interactive teaching of important cybersecurity topics.

Our Contribution. We present our approach to sensitize upper secondary school students (grades 10 to 12) for cybersecurity topics in a protected environment and build up basic knowledge about what happens when using different web services. MockHack contains an online learning environment, a *lesson center*, that is highly interactive. The students directly apply the content of the lessons on a *target application*, which mimics some real-world application the students interact with on a regular basis. *Helper tools* allow the students to inspect what happens behind the scenes when interacting with the target application; details are described in Section 3.

As a proof-of-concept, in Section 4 we present an instantiation of a target application that imitates a *social media network* offering basic functionality, e.g., to create and like posts. Two helper tools allow the students to observe network traffic, tinker with user authentication, and explore how certain social media features might actually be implemented behind the scenes; in particular, the cookie data can be inspected. The lesson center guides the students through the process, introduces terms like *request method* or *HTTP status code*, and finally enables the students to "hack" the social media network and impersonate another user.

In an initial evaluation, the feedback gathered from cybersecurity experts, teachers, as well as upper secondary school students was positive. We take this as a potential indicator that the platform is suitable for effective classroom teaching.

2 Related Work

Current State of Cyber Security Education. There is an increasing number of curricula frameworks and guidelines addressing cybersecurity topics, such as CS2023 [12], CyBOK [13], or e-CF [9]; these works have a scope broader than upper secondary school education, while the *K-12 Computer Science Framework* [4] and the *sIQurity K-12 curriculum* [17] are more narrow. Concrete implementation in classrooms seems to be behind in many countries, however. For instance, Stepney and Allison [18] studied K-12 curricula in England and found that "cyber security content is largely absent within secondary education curricula."

In [anonymized country], the situation is comparable. The recently published federal curriculum lists cybersecurity as a topic in mandatory upper secondary CS education, but without being very specific [anonymized], only vaguely describing competencies, e.g., when interacting with social media. Schools and teachers are tasked with deriving concrete learning goals and lesson plans from this high-level framework curriculum. However, many teachers do not feel well prepared; see Section 5.3.

Existing Tools. There is a large number of tools to educate users in cybersecurity topics, many of which use *gamification* such as *capture the flag* (CTF) approaches. *picoCTF* [19] is among the largest high-school hacking competitions as well as maintains a practice section (the *picoGym*) where challenges of various difficulties can be solved, including a hint system for users to receive further assistance when stuck. *Hack the box* [5] is a platform that provides a gamified hands-on learning experience, mainly tailored towards experienced hackers or professionals, but also provides a learning environment for universities. The *OverTheWire Wargames* [1] are a collection of gamified CTF challenges where each level on different paths built around different concepts (e.g., binary vulnerabilities or web security) requires solving the previous one. The *DOJO* [14] is an education platform developed by the Arizona State University providing a fully browser-based system using virtual desktops accessible directly through the web browser. The challenges are complemented by free lectures available on the website's YouTube channel with additional live streams through their Twitch channel. *Hacksplaining* [11] is a platform that introduces cyber threats such as SQL injections, cross-site scripting, etc. The US-based *National Initiative for Cybersecurity Careers and Studies* launched a *Cybersecurity Education and Training Assistance Program* [8]. One of the projects currently funded is *CYBER.ORG Range* [3], which allows to run virtual machines in the browser in a safe environment to learn hands-on about cyber threats.

With the exception of *picoCTF*, to the best of our knowledge, none of these platforms are tailored towards being used in K–12 classrooms, either focussing on college education or industry professionals. However, *picoCTF* focuses on education on the shell level, whereas MockHack focuses on the students' interaction with common web applications.

While MockHack is designed modularly to use different target applications for different lessons, we mainly look at the example of a social media network in this paper. A tool that follows a similar approach was presented by Hartl et al. [10], who introduced InstaClone, an Instagram-like mock platform that comes with a backend that allows the inspection of what data is stored in order to foster data literacy competencies; the focus is thus not on cybersecurity and the functionalities of the two tools differ significantly. Other noteworthy social media networks designed for educational purposes are TestDrive [6] and InstaHub [7] (and the recently introduced Opostum [16]); their focus is also not on cybersecurity.

3 Implementation and Usage

Problem Setting. As mentioned in Section 2, concrete approaches to teach cybersecurity in K-12 are in their infancy, both in [anonymized] and in a more general sense. Then again, in spite of a lack of teaching material and digital tools, curricula list cybersecurity as part of upper secondary school education.

Our goal was thus to enable educators to teach cybersecurity topics, from digital awareness to more involved technical concepts, to the target audience without the students having any advanced prior knowledge, meaning that the pre-knowledge conforms only to the minimum requirements of the [anonymized] upper secondary school curriculum.

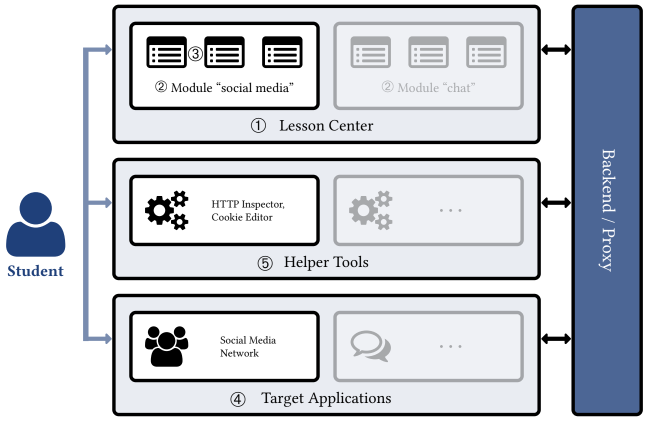


Figure 1: Students work through cybersecurity lessons with hands-on applied to interactive mock applications; progress is automatically tracked in a gamified environment.

Teacher Training. Our lab is responsible for teacher training in [anonymized]. Reaching out to our network of teachers, we conducted introductory MockHack workshops for around 100 teachers, with more workshops being currently scheduled.

Note that upper secondary teachers in [anonymized] have a Master's degree in CS and thus possess solid knowledge about cybersecurity topics. What they lack are teaching materials to teach these topics in a compelling and sustainable manner.

Deployment. MockHack is freely available at [anonymized] and only requires an email address to access. It contains *user* and *classroom management* functionality allowing teachers to setup an isolated instance for their class. The entry barrier is as low as possible:

(1) ***Teacher Account.*** Teachers contact our lab to create a *teacher account*. MockHack is hosted on our servers and usage is free.
(2) ***Course Setup.*** Teachers can then setup a *course*, i.e., an instance of MockHack for their class and invite students.
(3) ***Enrollment.*** Students register using a join code provided by the teacher, and work through the lessons at their own pace.

MockHack can be used in class as a complement to instruction phases, but it also allows an autodidactic approach: student accounts can be created independently, i.e., without joining a *course*.

System Overview. Figure 1 shows MockHack's design. Students interface with a lesson center ① where they can progress on different modules ②, each of which is composed of several lessons ③. These combine more traditional teaching techniques, e.g., explanations, with hands-on exercises to be solved in *target applications* ④: custom sandbox applications that mimic real services and allow students to apply the learned concepts. The students' progress is automatically tracked when they correctly solve exercises, providing an interactive and gamified environment. MockHack is fully browser-based and provides all the tools students need directly in the browser ⑤, eliminating the need for any additional software.

Integrating these target applications is essential for reinforcing cybersecurity concepts through practical application. MockHack provides a system for safe, controlled environments with artificial vulnerabilities tailored to each topic (e.g., a simulated social media network for safety lessons or a mock mailbox for phishing awareness). This ensures **students** can effectively apply and solidify their understanding in practical scenarios.

For **teachers**, MockHack provides a collection of ready-to-use lessons about different topics: these can be picked individually or in modules built around specific topics. Thanks to its browser-based nature, MockHack does not require any setup and supports a wide variety of devices. It offers convenient tools to enroll students into unique, isolated instances (*courses*), and provides monitoring and moderation tools for student supervision.

Components. In the following, we detail the main components of our system, namely the lesson center and the framework to support and monitor target applications and helper tools.

Lesson Center. The lesson center provides access to the different lessons, which can be optionally grouped together in modules and have pre-requirements, either enforced (e.g., having solved some other lessons) or informal (e.g., being familiar with a concept). A lesson consists of a sequence of lesson steps (Figure 2b), which can either teach some concepts, or ask students to apply their knowledge. This is done both through simple questions (e.g., single- or multiple-choice questions, or open questions that provide a place for the students to reflect and answer in free-text), and through tasks to perform on the *target application* that support the lesson.

As mentioned above, students can be grouped into *courses*, which is mainly beneficial for usage in classrooms. Teachers can create new courses, and students can join courses by entering the 6-digit join code that is displayed to the teacher.

Additionally, the system is configured such that students only see content generated on the target applications by other students of the same course, creating a closed system such that the students are safe in a classroom environment. We specifically decided against user-level isolation to keep some level of interactions between the students.

To avoid inappropriate content, teachers are able to moderate all content that is created on the target applications. We provide an example for our case study in Section 4.3.

Target Applications. The target applications form the accompanying playground for the students to apply their newly gained knowledge in a hands-on manner. They are small web-based applications that mimic their real-world equivalents. They may have specific vulnerabilities that the students are guided towards exploiting throughout the lessons.

Note that we want to preserve a safe learning environment, thus the system is built in a way that prevents exploiting these vulnerabilities outside the scope of the lessons. An example implementation is further discussed in Section 4.1.

Monitoring. All interactions with the target applications are sent through a proxy endpoint, which is a part of the lesson center backend and thus has access to all lesson data of the user making the request; see Figure 1. This allows for the implementation of a *live-update* feature where progress is automatically updated as soon as a task is fulfilled in the lesson's target application. From a technical perspective, the lesson center backend informs the frontend about a task completion through a WebSocket connection.

Helper Tools. Generally speaking, helper tools also mimic their real-life equivalents in a simplified manner; these are, e.g., analysis tools of some sort. As an example, we introduce an *HTTP inspector* in Section 4.2 that emulates the network traffic logs of the developer tools in most modern browsers. Helper tools aid with introducing core concepts without overwhelming novices with too many technical details and without requiring the installation of external tools, while providing a simple and easy-to-use design.

4 Case Study: Social Media and Internet

We now focus on a proof-of-concept instantiation of MockHack to teach students about web security and social media; see Figure 2. We address basic concepts of the web, such as client-server communication, HTTP, and cookies. Additionally, a goal is to raise awareness regarding online safety in social media.

The following sections discuss the individual components introduced in Section 3 that need instantiating, by first reviewing our implementation of a social media network including vulnerabilities, secondly explaining the helper tools that are used, and thirdly elaborating on how we designed the lessons.

4.1 Target Application: Social Media Platform

Our mock social media network supports the most well-known features of such applications, such as a *content creation system*, a *like system*, and a *follow system*. For the educational purpose of the application, two vulnerabilities have been implemented into it.

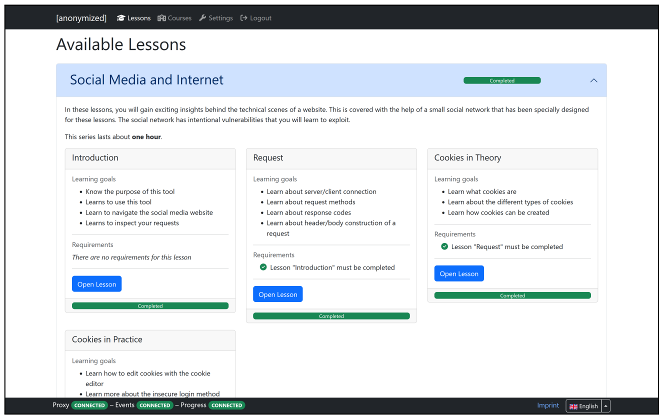
Vulnerability 1: Unsafe Authentication. The first vulnerability is of a technical nature: the social media network has an unsafe authentication mechanism. After providing the username and password to the backend when logging into the social media account, the server simply replies with a username cookie instead of anything that might be secure in the slightest. Any user is able to impersonate any other user by simply supplying the target's username in the username cookie. The students are guided towards this vulnerability step-by-step and will be able to understand and exploit it by the end of the lessons, as discussed further in Section 4.3.

The goal is to impersonate a specifically created account that is not associated with another real user (e.g., classmate). Impersonating real users is not possible; see below.

Vulnerability 2: Content Deletion. The second vulnerability is less of technical nature, and more designed to raise awareness about posting content online. When a user tries to delete any content they previously posted by clicking the red "delete" button, the content is not actually deleted on the server, but only a Boolean flag is set to true that indicates that the content should be considered deleted. It will then no longer show up on any user profile or timeline.

However, it is still accessible if a user possesses the direct link to that content, implying that users do not necessarily have control over "their" content after posting it.

Safety and Moderation. Certain restrictions are in place to ensure a safe environment for the students, especially with keeping a classroom deployment in mind. Even with the social media network having fundamental flaws in the authentication system, users are not able to access another real user's social media account. This is done by injecting some lesson center user info into all requests on the proxy endpoint. The social media network can then check if



Available Lessons

Social Media and Internet

In these lessons, you will gain exciting insights behind the technical scenes of a website. This is covered with the help of a small social network that has been specially designed for these lessons. The social network has intentional vulnerabilities that you will learn to exploit.

This series lasts about **one hour**.

Introduction

Learning goals
- Know the purpose of this tool
- Learns to use this tool
- Learn to navigate the social media website
- Learns to inspect your requests

Requirements
There are no requirements for this lesson

Request

Learning goals
- Learn about server/client connection
- Learn about request methods
- Learn about response codes
- Learn about header/body construction of a request

Requirements
✓ Lesson "Introduction" must be completed

Cookies in Theory

Learning goals
- Learn what cookies are
- Learn about the different types of cookies
- Learn how cookies can be created

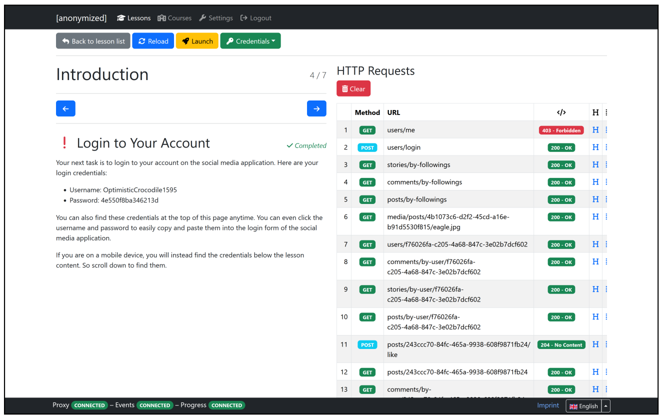
Requirements
✓ Lesson "Request" must be completed

Cookies in Practice

Learning goals
- Learn how to edit cookies with the cookie editor
- Learn more about the insecure login method

Proxy CONNECTED – Events CONNECTED – Progress CONNECTED

a: The lesson center showing four consecutive lessons



Introduction 4 / 7

! Login to Your Account ✓ *Completed*

Your next task is to login to your account on the social media application. Here are your login credentials:

- Username: OptimisticCrocodile1595
- Password: 4e550f8ba346213d

You can also find these credentials at the top of this page anytime. You can even click the username and password to easily copy and paste them into the login form of the social media application.

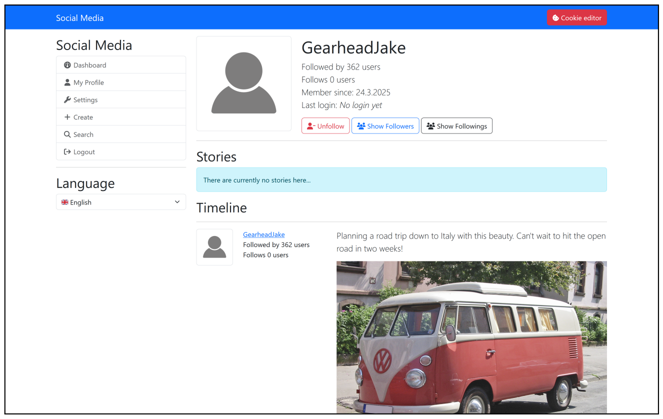
If you are on a mobile device, you will instead find the credentials below the lesson content. So scroll down to find them.

HTTP Requests

	Method	URL	</>	H
1	GET	users/me	403 - Forbidden	H
2	POST	users/login	200 - OK	H
3	GET	stories/by-followings	200 - OK	H
4	GET	comments/by-followings	200 - OK	H
5	GET	posts/by-followings	200 - OK	H
6	GET	media/posts/4b1073c6-d2f2-45cd-a16e-b91d5530f815/eagle.jpg	200 - OK	H
7	GET	users/f76026fa-c205-4a68-847c-3e02b7dcf602	200 - OK	H
8	GET	comments/by-user/f76026fa-c205-4a68-847c-3e02b7dcf602	200 - OK	H
9	GET	stories/by-user/f76026fa-c205-4a68-847c-3e02b7dcf602	200 - OK	H
10	GET	posts/by-user/f76026fa-c205-4a68-847c-3e02b7dcf602	200 - OK	H
11	POST	posts/243ccc70-84fc-465a-9938-608f9871fb24/like	204 - No Content	H
12	GET	posts/243ccc70-84fc-465a-9938-608f9871fb24	200 - OK	H
13	GET	comments/by-	200 - OK	H

Proxy CONNECTED – Events CONNECTED – Progress CONNECTED

b: A single lesson (left) and HTTP inspector (right)



Social Media

GearheadJake
Followed by 362 users
Follows 0 users
Member since: 24.3.2025
Last login: *No login yet*

Stories

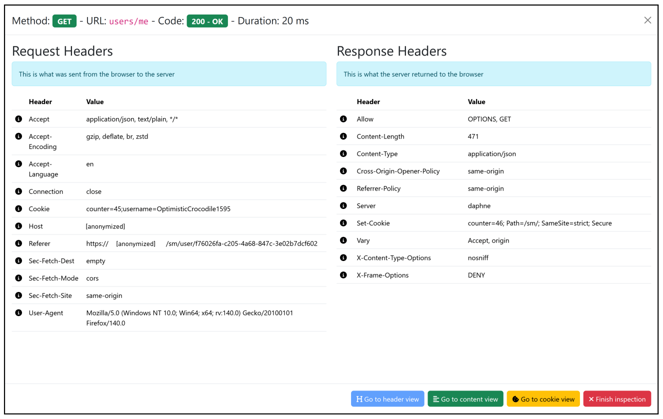
There are currently no stories here...

Timeline

GearheadJake
Followed by 362 users
Follows 0 users

Planning a road trip down to Italy with this beauty. Can't wait to hit the open road in two weeks!

c: The mock social media network (indicated by blue title line)



Method: GET - URL: `users/me` - Code: 200 - OK - Duration: 20 ms

Request Headers

This is what was sent from the browser to the server

Header	Value
Accept	application/json, text/plain, */*
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en
Connection	close
Cookie	counter=45;username=OptimisticCrocodile1595
Host	[anonymized]
Referer	https:// [anonymized] /sm/user/f76026fa-c205-4a68-847c-3e02b7dcf602
Sec-Fetch-Dest	empty
Sec-Fetch-Mode	cors
Sec-Fetch-Site	same-origin
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0

Response Headers

This is what the server returned to the browser

Header	Value
Allow	OPTIONS, GET
Content-Length	471
Content-Type	application/json
Cross-Origin-Opener-Policy	same-origin
Referrer-Policy	same-origin
Server	daphne
Set-Cookie	counter=46; Path=/sm/; SameSite=strict; Secure
Vary	Accept, origin
X-Content-Type-Options	nosniff
X-Frame-Options	DENY

d: The helper tool that allows to inspect headers

Figure 2: Screenshots depicting different parts of the platform

the user actually has access to the social media account they would like to access.

Another safety precaution is the introduction of a course sandbox. Users are only able to view content that was generated by users of the same course. For a classroom deployment, this means that users only see the content generated by their classmates. No outside content is visible, even if it exists.

To prevent inappropriate content, a moderation tool is implemented. Teachers have access to a special page that shows all content generated on the social media network in real time. If they detect any unwanted content, they are able to delete it on a single click. In contrast to the previously mentioned content deletion vulnerability, content that is deleted through this method will actually be removed from the social media network.

4.2 Helper Tools

We have implemented two specific helper tools to aid the students during the lessons. Firstly, we have created an *HTTP inspector* that lets the students examine the network traffic. Secondly, we have added a *cookie editor* that helps them with altering their cookies in a simple manner.

HTTP Inspector. A small HTTP inspector (Figure 2b) allows students to investigate the requests that are generated by their interaction with the social media network. Three different views are available: *header view*, *content view*, and *cookie view*. The header view shows all HTTP headers (Figure 2d); the content view displays the body comprehensibly formatted based on the content type: and the cookie view is a special view that only focuses on the `Cookie` and `Set-Cookie` headers.

The cookies that are sent are listed neatly, and the cookies that are received are displayed with all available information, such as cookie flags, or restrictions on domain or path. We decided to implement our own HTTP inspector instead of using the inspector directly present in most browsers, as the existing inspectors provide too much information for novices and might lead to cognitive overload.

Cookie Editor. Altering cookies using standard tools is quite a difficult task for novices, either requiring running commands in the JavaScript console, or other interactions with the developer tools. To not overwhelm the students, we have added a simple-to-use cookie editor to the social media network that lets students alter their cookies in a simple textual representation. The cookie editor can be opened by clicking the red button in the blue title line, as visible in Figure 2c.

4.3 Lesson Design

There are currently four lessons available addressing vulnerability 1 that are part of the module "Social Media and Internet." We introduce them in what follows; see Figure 2a.

Introduction. The first lesson is mainly introductory. The users are taught about the platform setup with the social media network as target application. They have small tasks, such as opening the target application and logging into their automatically created social media profile. As a final quiz, users are tasked to find the user ID of their social media profile by looking through the requests and using the HTTP inspector for the first time. Plenty of hints are provided and the whole lesson is very guided.

HTTP. The second lesson introduces a technical topic: HTTP. The students learn about certain aspects of this protocol with varying difficulty, ranging from introducing the client-server-communication paradigm to a detailed view of the transferred data, such as the request methods, the response codes, and the composition of headers and bodies. Additionally, they are introduced further into the HTTP inspector helper tool.

Cookies in Theory. The third lesson is also theoretical, introducing the theory behind HTTP cookies, as this is required knowledge for the final lesson. A small quiz is implemented at the end to probe the students' knowledge.

Cookies in Practice. The fourth lesson takes the learned theory and puts it into practice. As mentioned in Section 4.1, the social media network has an insecure authentication, as it just uses the username in a cookie. Users must now gain access to a target account and create a malicious post. They will achieve this by altering their cookies using the cookie editor introduced in Section 4.2.

5 Evaluation

The evaluation of MockHack has been performed in three phases. In the first phase, a group of ten cybersecurity experts was tasked with working through the existing lessons and providing subsequent feedback in semi-structured expert interviews. In the second phase, a pilot study was conducted at a [anonymized] high school. In the third phase, a survey was distributed among [anonymized] CS teachers. The IRB of [anonymized] approved our user studies.

5.1 Expert Testing

The initial evaluation phase of an early prototype of MockHack was a pre-pilot study where 10 experts worked through the lessons individually and at their own pace, while not being in a room together. All experts were members of the Computer Science Department of [anonymized], with six of them being scientific staff of the security groups. They were instructed to *think aloud*, allowing an observer to take notes. After finishing the lessons, each expert took part in a voluntary semi-structured interview, which was recorded and transcribed for analysis.

The notes taken while the experts were working through the lessons, together with the transcriptions of the semi-structured interviews, contributed a total of 89 individual points of feedback. The points which were mentioned by the most experts were prioritized, leading to an implementation of several usability improvements, including:

- Restructuring of the layout of the HTTP inspector;
- Implementation of live updates of the lesson progress;
- Usability improvements to the lesson structure; and
- Usability improvements to the social media network.

Regarding the lesson content, 9 experts believed that the introductory lesson effectively teaches how to use the platform. Similarly, 9 experts stated that the lesson on HTTP is technically accurate.

For the lesson on HTTP cookies, 7 experts felt that the discussion of the vulnerability would be engaging for students without a background in cybersecurity. However, only 6 experts thought that the concept of cookies was explained adequately, indicating a need for improvement. Furthermore, only 5 experts were confident that students could identify potential fixes for the broken authentication mechanism, suggesting that additional hints or guidance might be necessary.

Overall, 8 experts consistently knew what actions to take and what steps to perform. Similarly, 8 experts found the complexity of the helper tool to be appropriate. Generally, the experts were positive that the tool is useful for the target audience. One expert mentioned that they like that "it's sort of close to what you can actually see in the proper developer tools."

Nonetheless, some criticism was raised. One expert was concerned that the texts are too long, stating that "if you need to read 3 paragraphs to get the analogy, I would just skip this as a student." This might indicate that the texts can be designed more succinctly. Another expert already had suggestions for extensions, telling us that "what is definitely missing is HTTPS."

5.2 In-Class Testing

The second phase of the evaluation was a pilot study with upper secondary school students. 17 students of a single class participated in the study; they each had 90 minutes to work with MockHack individually. After that, they were asked to fill out a questionnaire which evaluates the score on the *System Usability Scale* (SUS) [2]. An average score of 72 (of at most 100) was achieved, which can be interpreted as "good usability" [15]. It should be noted that the results were in particular low for question 1 of the SUS ("I think that I would like to use this system frequently"), which is not very surprising at second glance. Indeed, our platform is not a tool for everyday student usage.

At the end of the questionnaire, the students were asked about their individual experiences and about what they liked and disliked most. With respect to the former, seven students (41 %) mentioned that they enjoyed working independently most. One student responded that they liked "getting things explained that you always use but never realize, such as cookies." Another student responded that "the system had good explanations with appropriate analogies. Quick start, learning with the system instead of too much theory."

Regarding what they disliked, multiple students mentioned that they think that the lessons are too text-heavy and could use more illustrations. Other students also responded that they wished that there were more complex tasks and that the theory could be more detailed and more in-depth.

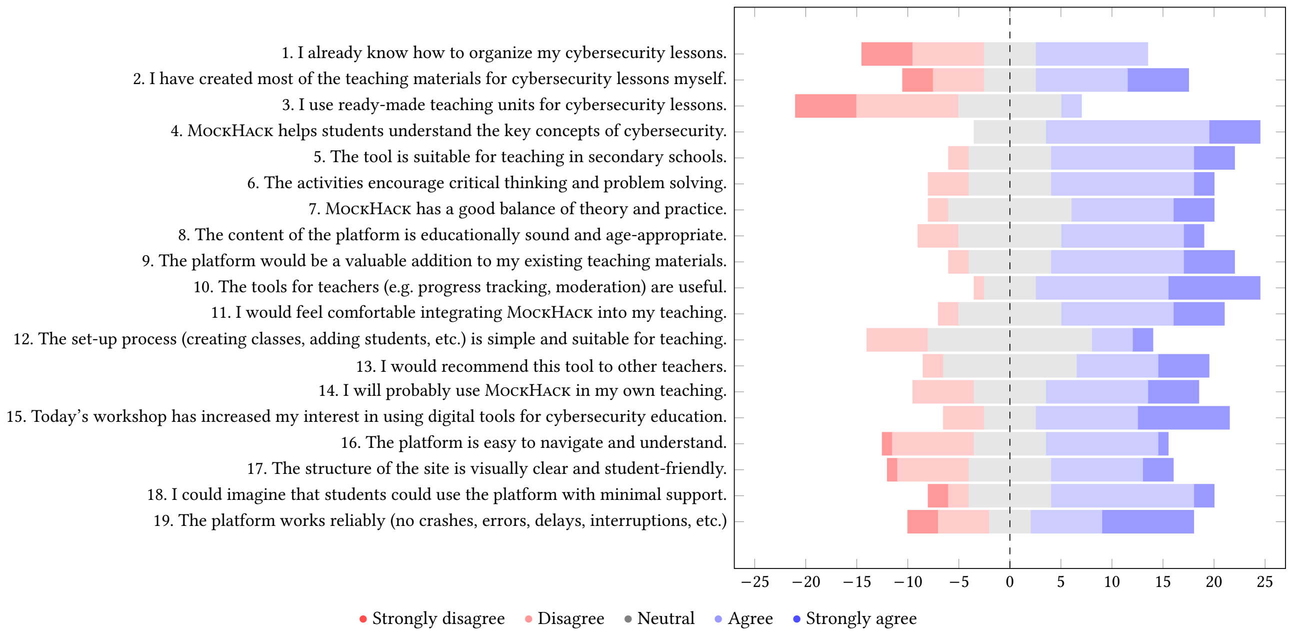


Figure 3: Accumulated answers of each item of the teacher survey

5.3 Teacher Survey

The third phase of the evaluation of MockHack was a survey that we distributed to teachers as part of a 90-minute workshop where they explored the tool from the viewpoint of a student. In total, 55 CS teachers took part in the workshop. Of these, 28 filled out the survey, which contained 19 items on a 5-point Likert-scale regarding educational benefits, integration into lessons, impressions, user-friendliness, and design. The results can be found in Figure 3.

Due to space constraints, we only mention the major takeaways. Particularly noteworthy are the teachers' answers to items 1–3, which hint at a large fraction of the teachers not being sufficiently prepared to teach cybersecurity topics in their classes. As mentioned in Section 2, while cybersecurity is part of the [anonymized] CS curriculum, there is no established teaching material. In turn, item 4, asking whether the teachers feel that MockHack would help their students, was agreed upon by 16 and strongly agreed upon by 5 of the 28 teachers. Furthermore, 8 teachers agreed and 5 agreed strongly that they would recommend the tool to other teachers (item 13). The answers to items 16–19 do however show that there is still room for improvement with respect to the platform design.

Additionally to the Likert items, teachers could provide free-text feedback. A lot of them expressed that they enjoyed working with the platform, and that they particularly liked the implementation of the realistic target applications as sandbox environments. Teachers also expressed their desire to adapt the (currently static) lesson content to their needs, as well as suggested improvements to the lesson structure and content.

6 Discussion

Our pilot studies provide evidence that MockHack can be used in upper secondary school education in order to address different aspects of cybersecurity, and the feedback from experts was overall very positive. There is some support that students can use it and indeed learn about the concepts taught in the individual lessons. The first findings using the SUS indicate that the design is generally appreciated. The remarks that the lessons "contain too much text" are (unfortunately) quite in line with what teachers report; it was remarked by an expert as well. We will assess whether the lessons can be broken down further, use more interactive elements, include short videos, etc. Besides that, feedback from teachers was generally positive, while we acknowledge that the next iteration has to improve the design of the tool.

We have so far only included one target application, the social media network. As indicated above, the design allows for far more flexibility. But also the current target application will be expanded further to teach students more about potential risks when interacting with social media.

7 Limitations and Conclusion

The evaluation of the tool has so far only been a first step. A more thorough analysis will investigate the students' learning gains, knowledge retention, as well as cognitive load. However, it was crucial for us to first find out whether teachers see a need for the tool and are willing to integrate it into their classes. Our preliminary findings suggest that this is the case for a significant fraction of the teachers.

The students' feedback shows a strong preference for the practical aspects of MockHack over the usual theoretical approaches. Moreover, their desire to delve even deeper into the topic indicates that we succeeded in providing a simple and accessible interface to an otherwise rather complex system. The mock social media network seems to be familiar enough to the students so that they perceive the overall learning outcome to be relevant. Further development and study of the platform is therefore well warranted.

References

[1] Steven Van Acker. 2024. OverTheWire: Wargames. https://overthewire.org/wargames/ Last accessed 20-Jan-2026.

[2] John Brooke. 1996. *SUS – A Quick and Dirty Usability Scale.* 189–194.

[3] Cyber Innovation Center. 2024. CYBER.ORG Range. https://cyber.org/range Last accessed 20-Jan-2026.

[4] K-12 Computer Science Framework Steering Committee. 2016. *K-12 Computer Science Framework.* Technical Report. New York, NY, USA. doi:10.1145/3079760

[5] Hack The Box Contributors. 2024. Hack The Box: Hacking Training For The Best – Individuals & Companies. https://www.hackthebox.com Last accessed 20-Jan-2026.

[6] Dominic DiFranzo, Yoon Hyung Choi, Amanda Purington, Jessie G. Taft, Janis Whitlock, and Natalya N. Bazarova. 2019. Social Media TestDrive: Real-World Social Media Education for the Next Generation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3290605.3300533

[7] Julian Dorn. 2019. InstaHub Datenbanken und Datenschutz mit einem extra für den Unterricht entwickelten sozialen Netzwerk unterrichten. In *Informatik für alle, 18. GI-Fachtagung Informatik und Schule, INFOS 2019, 16.-18. September 2019, Dortmund (LNI, Vol. P-288)*, Arno Pasternak (Ed.). Gesellschaft für Informatik, Bonn, 375. doi:10.18420/INFOS2019-F3

[8] National Initiative for Cybersecurity Careers and Studies. 2024. Cybersecurity Education and Training Assistance Program. https://niccs.cisa.gov/cybersecurity-career-resources/cybersecurity-education-and-training-assistance-program Last accessed 20-Jan-2026.

[9] CEN ICT Group. 2019. e-Competence Framework – IT Professionalism Europe. https://itprofessionalism.org/professionalism/e-competence-framework/ Last accessed 20-Jan-2026.

[10] Anna Hartl, Elena Starke, Angelina Voggenreiter, Doris Holzberger, Tilman Michaeli, and Jürgen Pfeffer. 2024. Empowering Digital Natives: InstaClone – A Novel Approach to Data Literacy Education in the Age of Social Media. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 484–490. doi:10.1145/3626252.3630839

[11] Hacksplaining Inc. 2024. Hacksplaining. https://www.hacksplaining.com/lessons/ Last accessed 20-Jan-2026.

[12] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023.* Association for Computing Machinery, New York, NY, USA. doi:10.1145/3664191

[13] Andrew Martin, Awais Rashid, Howard Chivers, Steve Schneider, Emil Lupu, and George Danezis. 2021. *Introduction to CyBOK Knowledge Area, Version 1.1.0.* The National Cyber Security Centre. https://www.cybok.org/media/downloads/Introduction_v1.1.0.pdf Last accessed 20-Jan-2026.

[14] Connor Nelson and Yan Shoshitaishvili. 2024. DOJO: Applied Cybersecurity Education in the Browser. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 930–936. doi:10.1145/3626252.3630836

[15] Jeff Sauro. 2011. Measuring Usability with the System Usability Scale. https://measuringu.com/sus/ Last accessed 20-Jan-2026.

[16] Thomas Schmalfeldt, Tobias Berner, Claudine Boyer, Mareike Düssel, Eike Rösch, Philipp Rüegerand, Moreno Kunz, Melanie Murer-Jäckle, and Sarah Hess. 2025. Opostum. https://opostum.ch/ Last accessed 20-Jan-2026.

[17] The sIQurity Foundation. 2026. K-12 Cybersecurity Curriculum | The sIQurity Foundation Programs. https://siqurity.org/pages/programs/k12-curriculum Last accessed 20-Jan-2026.

[18] Ollie Stepney and Jordan Allison. 2023. Cyber Security in English Secondary Education Curricula: A Preliminary Study. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 193–199. doi:10.1145/3545945.3569758

[19] Carnegie Mellon University. 2024. picoCTF – CMU Cybersecurity Competition. https://picoctf.org Last accessed 20-Jan-2026.