Capture-the-Flag challenges in higher education assignments

Influencing factors and requirements in development and user experience

Capture-the-Flag laborationer inom högre utbildning
Påverkande faktorer och krav i utveckling och användarupplevelse

Hugo Andersson
Per Andersson

Faculty of Health, Science and Technology

Computer Science

Bachelor Thesis 15hp

Supervisor: Leonardo Martucci

Examiner: Tobias Pulls

Date: 2023-05-31

Preface

Thanks to Leonardo Martucci, Jonathan Magnusson and Redpill-Linpro for your support throughout this project.

Abstract

Due to digitalization, computer systems have become integral to every aspect of our society. Not all the software and systems behind the wave of digitalization are securely developed, tested, or properly configured and are, therefore, vulnerable to attacks. The best way to protect ourselves is through increased awareness about these threats, where laboratory exercises are an excellent way to teach about the practical aspects of these things. To better understand what makes an excellent cyber security exercise, this thesis aims to develop a CTF-based laboratory exercise for the course Ethical Hacking at Karlstad University and analyze requirements and how different factors influence the development and user experience. To do this, we set up an environment for hosting, designed and implemented the exercises, and created questionnaires to gather participant data. As a result, we have created a list of 3 requirements and 4 critical factors together with an analysis of how they influence the development and user experience of the exercise. The most important results were that a correlation between difficulty and how much the participants liked the lab was found, questionnaire options should not be too broad since that makes the analysis of them less accurate, and distributing flags in web environments is more complex than we first assessed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today computer systems and software are present in almost every part of our society. Digitalization has not abated but will continue making computers a more and more integral part of our daily lives. Technologies such as computers, cell phones, and other tech that have come from the digitalization boom have great possibilities, but our dependency on them also makes us more vulnerable in case of an attack on these systems. Digitalization expands the possible attack vectors that malicious people and organizations can target which creates a dire need for increased awareness regarding cyber security throughout our society. One way to achieve this is by broadening the understanding of these issues among everyone who works in the IT industry. Concepts in computer science can rarely be fully understood from solely reading about them, they also need to be experienced firsthand in practice. An efficient way to acquire practical knowledge in computer science is through laboratory assignments. In this thesis, we have created a laboratory assignment in cyber security and analyzed what factors influence the development and user experience. The assignment was developed for the course Ethical Hacking at Karlstad University in the form of a Capture The Flag (CTF) [1].

1.1 Problem description

In this thesis we aim to answer the question “How do different factors and requirements influence development and user experience in CTF-based laboratory assignments?”

1.2 Goal and Purpose

The goal of this thesis is to investigate how requirements and factors impact development and user experience in cyber security-based laboratory exercises. These requirements and factors will be evaluated based on data gathered from the CTF that we will design and implement. This CTF will be part of Karlstad Universitys Ethical Hacking course DVAD25, where the purpose is to teach the participants about a specific vulnerability, the target group for the assignment is developers with some prior knowledge in cyber security. To achieve this we needed to create a laboratory assignment, host it, generate data from it, and analyze the data. The data analysis will help us determine which requirements and factor influences user experience and how.

1.3 Ethical Issues & Considerations

The ethical dilemmas that we need to take into consideration when creating this laboratory assignment is that the participants will be taught how to exploit vulnerabilities, that can be taken advantage of to perform both defensive and offensive actions. Our goal is to shed light on and educate the participants about the vulnerabilities so that they in the future can develop more secure software that these actions can’t exploit. How the participants use the information however is in the end up to themselves [2].

This leads to an even bigger ethical question, should Ethical Hacking be taught at all? Since educators can’t know student’s true intentions it might be counterproductive to educate students on how to exploit vulnerabilities that can be used for unlawful hacking. Jamil and Khan [2] presents the quote “A problem with teaching undergraduate students using this approach is that the instructor is effectively providing them with a loaded gun”. This quote shines a light on how dangerous it might be to educate students about intrusion/hacking. The educators give students tools, “how to” guides in intrusion and potentially lays the foundation for a malicious career in hacking. The message behind the quote is hard to argue against. Educators today are giving students knowledge about things that might be detrimental to society. We do not believe that the solution is to remove these “loaded guns”, since if that were the case one could make the same argument for disarming the police and military. The police and the army need to be armed in order to protect our society, just as a software developer needs knowledge on how to protect their and society’s assets from actors with bad intentions. And yes, stopping education about intrusion would probably mean fewer malicious actors with a lot of firepower but it would also weaken “the good guys” in this scenario. We believe however that the potential of students writing more secure code by being aware of potential intrusion threats outweighs the risk of them using it for malicious intentions [2].

Another ethical consideration in this thesis is the fact that we are collecting data about the participants. This data is being collected throughout the laboratory exercise. To our best abilities, we tried to give the participants as much integrity as we could by anonymizing the data, providing the right to revoke the data and the use of a consent form which can be seen in Appendix B. We believe that these precautions we took are sufficient enough for this thesis given that we do not collect data that is, in our opinion, very sensitive.

One last dilemma we took into consideration was that the students were offered extra points in the course DVAD25 if they finished the laboratory exercises before the course ended. The dilemma is that some students may feel forced to complete our laboratory assignment to get a higher score on the course. One might also argue that it might be unfair for students that do not have enough points in the course and need the points from our laboratory assignment to pass, since we have no experience in creating laboratory assignments it might be badly structured, hard to understand, and so on. On the other hand, one can argue that for this laboratory assignment, the students only get extra points, with an emphasis on extra. We needed to make a decision on whether the students should be able to decline our consent form and still be able to complete the lab and get the extra points in the course. We decided to not let them do this since we did not want to risk not getting enough data for our analysis. If we had more participants we might have reasoned differently.

1.4 Method

As mentioned in section 1.2, the goal of this thesis is to investigate how requirements and factors impact development and user experience in cyber security-based laboratory exercises. To successfully achieve this goal we researched vulnerabilities, planned time allocation and selection of data points to gather for analysis, designed and implemented exercises based on the vulnerabilities, used a test group for evaluation of implementation, ran the laboratory assignment over a four-week time period with a main group and gathered data from the exercises, evaluated and analyzed the data.

1.5 Stakeholder

We did our thesis at the IT consultant company Redpill-Linpro, but the project stakeholder is Karlstad University. The project was part of the course DVAD25 in spring 2023 and might be used as an exercise for the course in the future to further deepen student's knowledge about the vulnerability XSS. XSS is a vulnerability that has not previously been included in the course, and might therefore benefit future students. The reason that we wanted to do our thesis at a company even though the stakeholder is Karlstads University is to expand our professional network, expand our knowledge on how it is to work as a consultant in the IT industry, and get to know a possible future employer.

1.6 Work Distribution

This section has been divided into three sections. One for the tasks we did together and one section each for our individual work.

Hugo

For this project, Hugo Andersson had the responsibility to set up the server to host the CTFd container and the server for the container running the vulnerable website. The server setup also includes configurations such as file distribution, user management, and more. He also created a bot for the last exercise to simulate a victim visiting the URL, which included researching Python modules such as requests and selenium.

Per

In this project, Per Andersson configured the CTFd container, added plugins, researched how to best execute the containers needed, designed and created two of the exercises for the CTF, and wrote code to prevent users from accessing the flags from the browser through inspect or by viewing the source code.

Joint Collaboration

Together we came up with requirements and factors for creating a good CTF and researched the different possible vulnerabilities that we could use as a topic for the challenge. We then considered both pros and cons of the vulnerabilities discussed and decided which topic to choose. We also designed the questionnaires that the users would need to answer together. At the end of the project, we also analyzed the data generated by the participants.

In the data collection and analysis, we did not distribute the work, but instead worked together. We did this since data collection and analysis is rather complex which makes it easy to do mistakes that will lead to an inaccurate and incorrect result.

Apart from writing separately about the implementations we wrote and proofread the thesis together.

1.7 Limitations

The limitations of this thesis are presented in the following list:

- In this thesis, we implement four CTF challenges that each teach the students about a certain security flaw regarding cross-site scripting (XSS) attacks. There are myriads of attack vectors in web applications however meaning that securing these does not guarantee in any way that the site is perfectly safe since there could be other attack vectors open for an attacker. For example, stored XSS has been excluded from this thesis completely. This does not answer queries such as; “Is website X secure if the methods used in this thesis can’t be used to crack it?”
- The relatively small number of participants that we have gathered data from makes us unable to declare any statistical significance. We have however used the data as indications to support our thoughts and arguments.
- To further understand what factors influence development and user experience it would have been more optimal to have the participants complete several CTF’s that were designed in different ways, focusing on different vulnerabilities and

fields of computer science to enable us to cross-reference our findings and conclusions.

- Questionnaires have their limitations since they are subjective. The participants might for example feel some prestige and report a lower difficulty or time consumption than they actually felt or used.

1.8 Disposition

The background chapter 2 gives an extensive description of each major technology used in the implementation part of this thesis work. The design chapter 3 presents how we chose the topic of the CTF by coming up with different requirements and factors. Chapter 4 presents how the CTF has been implemented. In this chapter, we go into detail on how each of the different technologies explained in Chapter 2 has been used to create the CTF for this thesis. In chapter 5, we discuss and analyze the results from the questionnaires together with our own experiences from the development of the project. In the Conclusion chapter 6 we will discuss the project outcome and future work.

Chapter 2

Background

The laboratory assignment was in the form of a CTF which is a concept used for constructing cybersecurity challenges. In a CTF participants will be given a starting point and their task is to find a hidden flag. There are different types of CTF's, some are for competitive purposes and some are for educational purposes. There is also the difference between attack/defense-style CTF's, where the participants steal the flag from each other, and jeopardy-style CTF's, where participants steal flags from the CTF organizer [3]. The starting point can be anything from a URL, file, local or remote machines, and more. The flag can be anything but it is a form of verification that the CTF participant has reached a certain point or achieved a given goal. More often than not the flag is a string in the format FLAG{substring}, where the substring is any string of characters unknown to the participant. There are no limitations on how the participants acquire the flag, meaning they can use any methods, tools, and knowledge to acquire it [4]. To implement the CTF there were several technologies necessary to make everything work. To create the assignment there were several required functionalities. These functionalities are presented in the following list:

- A platform that could handle scorekeeping, validation of flags, user registration and oversight, distribution of challenges, and data handling.

- A vulnerability to base the exercises on.
- A time-efficient deployment environment that was easy to use.
- A viable solution for hosting the exercises, platform, and other related components in need of hosting.
- An approach to emulate user interactions on a web page.
- An approach to handling and analyzing data.

Chapter 2 is structured as follows; In section 2.1, an explanation is provided regarding the CTFd platform. Section 2.2 provides an explanation of the chosen vulnerability XSS. An explanation of Docker, Docker images and containers, and Docker-Compose is presented in section 2.3. In section 2.4 a summary of VULTR and the functionality used in this project is presented. The Python package selenium is presented in section 2.5. The second to last section 2.6 summarizes the functionalities used in the Python pandas package used for this project. The last section 2.7, provides a summary of the functionalities utilized in this project from the Python Seaborn package.

2.1 CTFd

We used CTFd as the environment to host the CTF. CTFd is an open-source CTF platform that includes functionalities like a presentation of exercises, reporting of flags, user registration, score system, admin interface, and a database to store a broad variety of data generated from the users, for example, the number of attempts and submissions. CTFd has an easy-to-use interface where new tasks get unlocked as the user finishes questionnaires and exercises, as is shown in Figure 2.1.

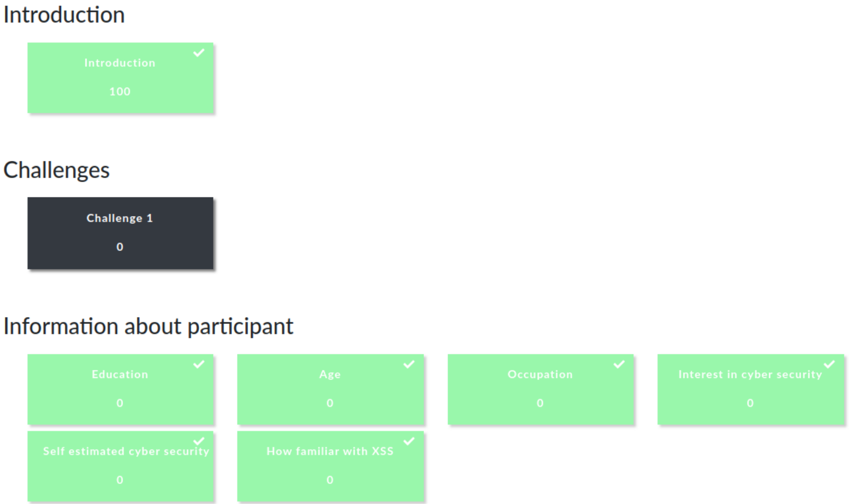
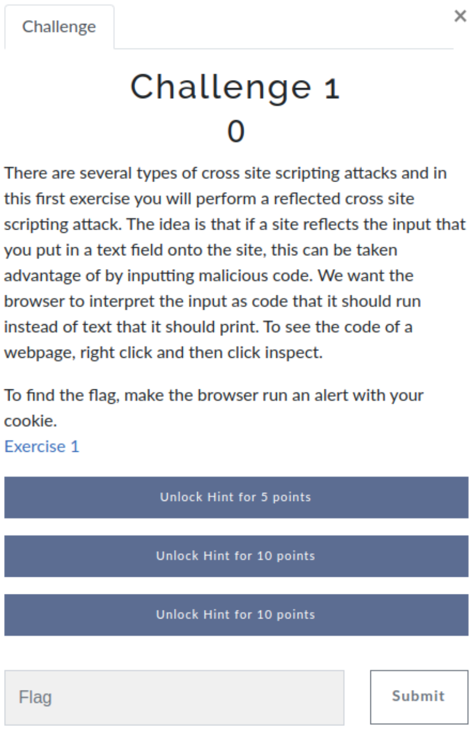


Figure 2.1: CTFd Challenge Overview

When clicking on a challenge a window will pop up with some text and an input field, as shown in Figure 2.2. The user also gets the option to buy hints. These hints will however reduce the total amount of points rewarded to the participant for solving the exercise.



Challenge 1

0

There are several types of cross site scripting attacks and in this first exercise you will perform a reflected cross site scripting attack. The idea is that if a site reflects the input that you put in a text field onto the site, this can be taken advantage of by inputting malicious code. We want the browser to interpret the input as code that it should run instead of text that it should print. To see the code of a webpage, right click and then click inspect.

To find the flag, make the browser run an alert with your cookie.

Exercise 1

Figure 2.2: CTFd Challenge Description

The CTFd platform was the platform of choice for this project because it fulfilled all the requirements that were presented in list 2. Another reason for choosing CTFd was because of the possibility to host it locally.

2.2 Cross Site Scripting (XSS)

XSS is an attack where the attacker tries to inject scripts into web applications. When the application is viewed by other users the scripts will execute on the participant's browser. There exists a variety of different XSS attacks since there are many different attack vectors in web applications. Some examples of attack vectors are input fields, URLs, and databases. The type of web-specific attacks that this thesis will touch upon are reflected XSS and Document Object Model (DOM) XSS [5].

Reflected XSS means that input sent to the web server and then parsed will be interpreted as code. When the input is interpreted as code the web server will execute that code and send/reflect the result of the execution back to the web page. An example of this would be that when a user searches for “cucumber” the string “Sorry cucumber was not found” is reflected to the web page. This reflection of the input field to the web page is then taken advantage of to make the browser run code that might be malicious. The first step is to break out of the code that makes the browser interpret it as something that should be rendered on the website and make it interpret it as code that it should execute. After this is done the attacker can usually execute any commands by injecting them [5].

The next step in reflected XSS is to make the victim visit a site and instead of the result being reflected in the victim’s browser the result is instead sent to a server that has been set up by an attacker. This type of reflected XSS works by the creation of a link with a malicious payload. This payload often consists of a script constructed to get information about the victim, for example, cookies, and also a component that points to the attacker’s server. When the victim clicks the link it sends the script with the malicious code to the web server and the result is reflected to the attacker’s server. To make the victim press the link it is usually sent with a phishing email. Phishing means that the attacker is trying to impersonate a legitimate source with the purpose to make the victim press the link, for example, a government agency or a lottery company. The good thing about this attack is that if the victim is vigilant, and doesn’t press the link, the attack can be prevented. This attack can also be prevented from the server side by sanitizing the input

DOM XSS attacks are similar to reflected XSS in the way that the attack is usually sent with a malicious link and email, but it instead targets the DOM. DOM is a web API that is used to build websites. An example of a DOM attack vector is the environment variables in the site's URL. An example of an environment variable could be language=french in "www.mysite.se?language=french". These kinds of attacks can contrary to reflected XSS be used when there are no input fields to target. These attacks can be prevented similarly as for reflected XSS [5].

2.3 Docker

Docker is a virtualization technology for containers. The main purpose of docker is to provide developers with a solution to build, ship, and run applications in any host that runs docker. Because docker provides easy setup and configuration to applications along with dependency handling it decreases the time needed for deployment [6]. Since meeting deadlines is an important requirement in this project, the choice to use docker is justified by the time saved in deployment.

Docker offers a lot of different components for different purposes. For the purpose of this project, it is important to understand the Docker components: Docker Images, Docker Containers, and Docker Compose. These will be explained in the following sections.

Docker Images & Containers

In docker two major components are often mentioned, these are docker images and docker containers. An image is a system specification and the container is the software that runs the system. The applications built in docker are packaged with all the dependencies into a container [7].

A container is a process that has been sand-boxed on a machine. The container is isolated so it won't interfere with other running processes on the machine. A container packages code and all its dependencies so the application can run quickly and reliably from one computing environment to another [8].

Docker-Compose

A lot of the software systems built today are multi-component systems. This is the main use case of Docker Compose (DC). According to docker [9] “Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your applications services. Then, with a single command, you create and start all the services from your configuration.” This YAML [1] file that docker is specifying is the key component of DC. When the DC command is executed it will look for the compose YAML file, in this file the developer defines a system by specifying the:

- Services
- Networks
- Secrets
- Volumes
- Config

It is shown in Appendix Awhat a docker-compose file might look like, but the basic concept is that the developer specifies relationships between containers, how they depend on each other, and how they work together. The developer also specifies how to build the containers and how the configuration of each container should be when the system executes [11].

[1] YAML is a data-serialization language, it is often used in configuration files and data storage [10]

2.4 VULTR

VULTR is a virtual Private Server provider (VPS)[2] that provide customers with resources to perform cloud computing. According to VULTR [13], they are a cloud infrastructure provider that provides various products such as VPS server hosting to customers [13]. The main reason for choosing VULTR as the server provider for this project was because of the low time it took to deploy servers and the low cost of server hosting.

2.5 Selenium WebDriver

Selenium WebDriver is an API that provides an interface to control web browsers without any human interaction, such as emulating user interactions, reading HTML, and more. Selenium's intended use is to perform automated tests on websites, but because of the functionality it provides it's also an excellent framework to use when building bots meant to simulate user interactions. Appendix B shows how the selenium WebDriver framework can be used to create a bot that sets the cookie to a specified string and then takes a URL, checks if it is valid, and visits it for the purpose of providing the cookie to an external source[14].

2.6 Pandas

Pandas is a framework based on numpy[3]in python used to structure/re-structure data, from formats such as JSON or CSV into data structures called panda data frames or in some cases data series. The advantage of these data frames is that they make it easier

[2] A VPS is an isolated virtual environment based on a physical server. VPS uses virtualization to divide physical machines into several theoretical machines.[12]

[3] Numpy is a python package used for scientific computing regarding multidimensional array objects [15]

to work with, understand and visually present the data. Pandas are used for labeled data which means that every data point in a set has a description or can be categorized by some attributes that the data has. The data frame has a two-dimensional column-row-based approach to structure the data, where each column corresponds to an attribute and each row represents an actual value/data point, this structure draws many similarities to both Excel spreadsheets and SQL tables [16]. Pandas were chosen as the data handling framework for this project because of certain functionalities it provides that suit the requirement.

2.7 Seaborn

Seaborn is a python framework built on top of Matplotlib[4]. The Seaborn framework is used to visualize statistical data in Python by giving the developer an easy-to-use API for the Matplotlib functionality with the intent to relieve the developer of needing to know how to draw each graph but instead, focus on the difference/importance of the data. Seaborn visualization comes in the form of graphs such as histograms, scatter plots, and many other types. Seaborn is closely connected with pandas as pandas are the main data structure that Seaborn "wants" to work with, therefore Seaborn was chosen as the data analyzation/visualization tool for this project [18].

[4] Matplotlib is a python package used for creating different types of visualizations [17].

Chapter 3

Design

When designing the exercises, the university gave us free reins regarding what security topic to base the laboratory assignment on and how to design and implement it as long as the requirements were met. In addition to the assignment being in the form of a CTF, the requirements were:

- Enough time to implement the exercises
- Topic not covered by other, preexisting laboratory assignments
- Success evaluation and flag distribution

In addition to the requirements, we wanted to analyze which other factors would influence the outcome of the assignment. We researched briefly within the scope of cybersecurity academic papers regarding factors influencing laboratory assignments and found limited literature on this topic. This research could have been more thorough but because of the time constraint, the decision was made to not continue our research. Instead, we talked to our supervisor and brainstormed possible factors. We came up with four factors that we believed would influence the outcome of the assignment. We talked to the stakeholder who agreed that these factors would be of interest to research in this thesis. The factors that we decided on were:

- Time for the participants to solve the exercises
- Relevance for the participants
- How engaging the exercises would be for the participants
- Experienced difficulty

In this thesis, we will focus on analyzing these requirements and factors.

3.1 Requirements

In this section, we will discuss the justification for each requirement.

Enough time to implement the exercises

Time estimates tend to be complicated in software development. Most software development projects have a deadline which makes it important to be able to estimate the expected time consumption for the project. During this project, we had a strict deadline for when the implementation of the laboratory assignment. This was because the students in the course DVAD25 needed to have enough time to finish the assignment before the course ended for them to earn extra credit points. We needed to meet this deadline since solving it after the course ended would not contribute to their grade. This would have reduced the incentive for the students to finish it, which would result in less data for us to analyze.

Topic not covered by other, preexisting laboratory assignments

Since there is no point in developing an assignment that covers topics or has the same learning outcome as assignments that already exists in the course, a requirement from the university was that the topic was not already covered by other assignments.

Distribution of flags and success evaluation

The requirement distribution of flags and success evaluation means finding a way to give participants the flag when they have reached a certain goal. This was required since it was a necessity for us to be able to determine if the participants completed the exercise or not. We needed to make sure that the participants did not exploit our system/implementation to acquire flags in a way that was not intended.

3.2 Factors

In this section, it will be discussed why the factors we choose are important when designing CTF-based laboratory assignments.

Time for the participants to solve the exercises

Time for participants to solve the exercises is an important factor since if it is too long many participants might lose interest and not finish, which would result in less data for us to use for our analysis. A short assignment would not allow the laboratory assignment to go into enough depth in the chosen topic, and therefore not give the participants a good learning outcome.

Relevance for the participants

When designing a laboratory assignment the relevance for the students is an important factor. Vulnerabilities that are common and realistic have a higher relevance than vulnerabilities that do not. For example, buffer overflow attacks require a lot of safety mechanisms to be disabled in order to make them possible. This makes them less relevant and realistic [19].

How engaging the exercises would be for the participants

When creating a laboratory assignment, it needs to be taken into consideration how interesting and engaging the user experience will be. An exercise that the participants find unappealing will make it less likely that they complete the assignment, which makes this factor important. How engaging something is, is very subjective, which makes this factor hard to analyze.

Experienced Difficulty

The level of difficulty is a crucial factor to consider since an excessively challenging assignment may overwhelm students and impede their ability to complete it, while an excessively simplistic assignment may not provide the fullest extent of learning opportunities for the participants.

3.3 Choosing vulnerability

After listing the requirements and the factors we decided which vulnerability to implement. For the decision we first did an overview on possible vulnerabilities to implement. We found seven vulnerabilities that we had enough prior knowledge of to be confident in spending time performing a deeper analysis. The vulnerabilities we considered were:

- Buffer overflow
- Social engineering
- Malware
- Reverse engineering
- Binary exploit
- XSS

XSS

The decision was made to implement an XSS vulnerability. This was because XSS fulfilled all the requirements previously mentioned in this chapter and seemed to enable us to get a decent outcome in the most important factors, also mentioned in this chapter. The decision not to use the other vulnerabilities was because of various reasons which do not impact this thesis. We will in this section discuss why we choose XSS with regard to the requirements and the factors.

Enough time to implement the exercises

To create an XSS exercise we knew beforehand that we needed to set up a website with vulnerabilities. Although web development is a new field for us we assessed that setting up a basic web page would not be difficult or time-consuming given that we have experience as developers.

Topic not covered by other, preexisting laboratory assignments

Since our laboratory assignment was going to be part of a course at Karlstad University, we knew which prerequisites the course had. This in combination with a list of all other laboratory assignments included in the course DVAD25 helped us determine if there would be an overlap or not, and if so, how big. There were no other exercises on XSS in DVAD25. However, the course DVGC19, which is a prerequisite to DVAD25 had one assignment on XSS. We believed that the XSS exercise in DVGC19 was not exploring the topic thoroughly, and because of this we reasoned that it was a relevant vulnerability to implement.

Flag distribution and success evaluation

Since XSS is a web-based vulnerability we reasoned that it would make it easy to distribute flags. The rationale behind our decision was based on the knowledge that the same approach had been previously employed, for example in Hack The Box's CTF on XSS [20].

Relevance for the participants

With XSS we would be able to create an assignment that was close to reality, therefore making it relevant. When designing the websites for XSS it is necessary to exclude safety measures related for educational purposes. However, it is not unreasonable to exclude these safety measures since XSS is one of the most common attacks and the safety measures often originate from actual XSS attacks. Far from all web developers takes all the safety measures into consideration. It is not only important to learn about the attack to build better web applications but also to protect oneself from these attacks [21].

How engaging the exercises would be for the participants

Since XSS is web-based we reasoned that it would give us a lot of freedom in regards to the design. Because of the freedom in the design, we had good possibilities to make it engaging. Using a web environment enabled us to make it interactive and closely related to how XSS attacks function in real situations, we reasoned that this would make the exercises more engaging.

Time for participants to solve the exercises

XSS is a vulnerability that is relatively easy to get started with as it does not demand extensive expertise beyond general programming knowledge. Because of the level of expertise needed to get started with XSS, we reasoned that it would be easy to implement challenges that are not excessively time-consuming.

Difficulty

We believed XSS would be a good vulnerability in regard to difficulty. This is because the XSS vulnerabilities can be designed in a myriad of ways, with a wide range of difficulty levels.

3.4 Designing the XSS laboratory exercise

After the decision to base the assignment on XSS, we did thorough research on the topic. There are several subcategories of XSS attack where the most common ones are reflected, DOM, and stored XSS [5]. Reflected and DOM XSS only requires a web page while stored XSS in addition requires a database. Since we had a strict deadline we decided to only include reflected and DOM XSS attacks in our laboratory exercise. After consideration in regards to the deadline, it was determined that the implementation should consist of four exercises. We decided that we wanted two websites with input fields, one site without any input so that the participants will need to attack other components of the site, and one site used for interactions with our bot. See implementation chapter 4 for further details.

To be able to analyze the factors and answer the thesis, it is not necessary for the assignment we created to have good scores in the factors we decided on. A laboratory assignment with low scores might have given us equally analyzable data as an assignment with high scores since the participants pointing out what is negative with the assignment could be used to help us understand what makes a good assignment. In addition, what we think are good factors for a laboratory assignment might be the opposite and vice versa. Nevertheless, we want as many students to finish the assignment as possible, to get a lot of data. It was needed to take into consideration that the laboratory assignment was going to be a part of the course DVAD25. With the influence of the student's grade, it is important that the assignment is professionally made and contributes to the course goals. With these arguments in mind, we decided to design a challenge that would score high in the factors from the design phase.

Consent Form

A requirement from the university was that we created a consent form for the participants to accept before using their data in our analysis and a right to revoke the consent functionality.

Participants

To make the implementation iterative we decided to use a test group and a main group. The idea behind the test group was to use a small number of participants for the evaluation of the assignment. Their feedback was going to be used for adjustments before releasing the assignment to the main group. The initial thought behind the main group was to use students from the course DVAD25, but after consideration, the decision was made to also include employees from Redpill-Linpro in the main group. The choice of adding the employees to the main group was to generate more data.

Data Gathering

To perform an analysis of the factors we decided to analyze, it was necessary to collect data from the participants. Initially, we considered monitoring the participant's interactions with the server by logging them. After some consideration, we determined that implementing such a logging system would require a significant amount of time, time that would be better spent developing exercises. Therefore, we decided to use questionnaires to gather data from the participants. These questionnaires were designed to be completed by the participants as they worked on the assignment. When designing the questions our goal was to cover as many relevant data points as possible. The questionnaire questions along with given alternatives will be presented in the following table 3.1 and list.

In table 3.1 the following words has been abbreviated:

- Master in Computer Engineering → MCE
- Bachelor in Computer Engineering → BCE
- Bachelor in Computer Science → BCS
- Cyber Security → CS

Table 3.1: Questionnaire Given Alternatives

Question:	Answer 1	Answer 2	Answer 3	Answer 4	Answer
Education	Master in CE	Bachelor in CE	Bachelor in CS	Other	-
Age	18-23	24-29	30-35	36-41	42+
Occupation	Student	Teacher	Employee	Other	-
Interest in CS	1 (Not at all)	2	3	4	5 (Alot)
Self estimated CS skills	1 (Low)	2	3	4	5 (High)
How familiar with XSS	1 (Not at all)	2	3	4	5 (Alot)
Time spent on challenge 1	0-10	11-20	21-30	31-60	60+
Difficulty challenge 1	1 (Easy)	2	3	4	5 (Hard)
Time spent on challenge 2	0-10	11-20	21-30	31-60	60+
Difficulty challenge 2	1 (Easy)	2	3	4	5 (Hard)
Time spent on challenge 3	0-10	11-20	21-30	31-60	60+
Difficulty challenge 3	1 (Easy)	2	3	4	5 (Hard)
Time spent on challenge 4	0-10	11-20	21-30	31-60	60+
Difficulty challenge 4	1 (Easy)	2	3	4	5 (Hard)
Did you like the lab?	1 (Not at all)	2	3	4	5 (Alot)
Self estimated WB skills	1 (Low)	2	3	4	5 (High)

The following list presents the open-answer questions.

- Test environment, Did you cheat/ break our test environment to complete the exercise?(There will be no repercussions if you answer yes) If yes how did you do it?
- What do you think could be improved?
- Other feedback

Chapter 4

Implementation

For this project, a variety of technologies was used, as shown in Chapter 2. This Chapter 4 has been organized, to separate the implementation, as follows: CTFd, Websites, Hosting, Bot and Pandas, and Seaborn. All the code can be found at `https://github.com/Redishh/CTF-LaboratoryAssignment`.

4.1 CTFd

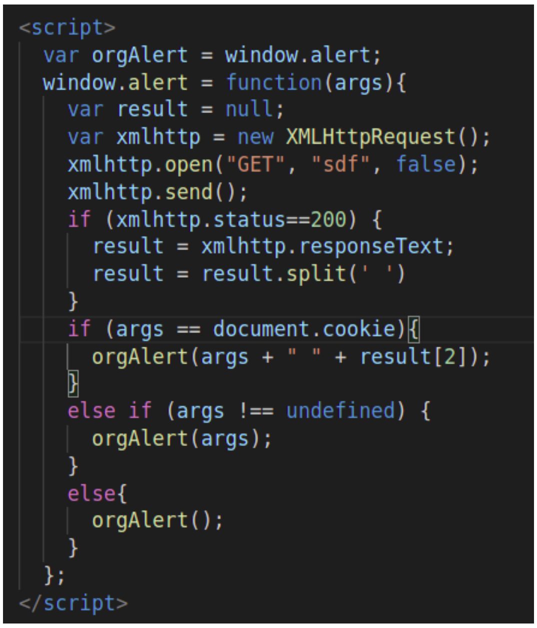
The CTFd platform used for this project is a docker image retrieved from docker-hub[1]. This image ships with a standard configuration for CTFd, but this was not enough for our intended use of CTFd regarding functionality. We wanted to add functionality for questionnaires on the platform since that would make it easier for the participants to answer our questions, allowing them to click on their answers instead of typing them. This required additional configuration on top of the existing docker image. Since CTFd is an open-source platform we could have designed it ourselves, but after some consideration, we bought a plugin instead with the functionality we wanted. We decided to use the plugin because it would have taken valuable time to implement the

[1] Docker Hub is the worlds largest repository of container images[22]

functionality on our own, time that would be better spent on developing the actual CTF exercises. The rest of the work with the platform was about adding our questionnaires and challenges through the graphical interface.

4.2 Websites

The Web pages are built with PHP, HTML, and JavaScript. The web pages have a simple design but there were a few things that we had to solve for the exercises to work in regards to flag distribution. We wanted the participant to acquire the flags only when they executed a JavaScript alert with the cookie passed as a parameter. Since the participant's cookie was the flag we did not want them to be able to find the flag/cookie through their browser menu. To solve this we overrode the alert function and added the flag only when an alert was executed with the participant's cookie. We realized that the users could see the flags if they inspected the web page source code so we added another layer of obfuscation by reading the flags from a file which is shown in Figure 4.1.

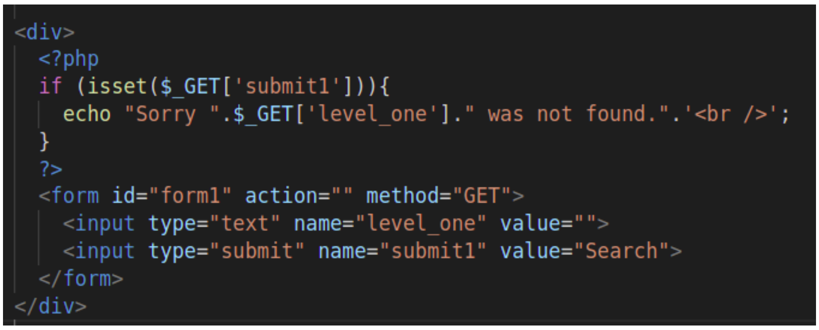


```
<script>
  var orgAlert = window.alert;
  window.alert = function(args){
    var result = null;
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("GET", "sdf", false);
    xmlhttp.send();
    if (xmlhttp.status==200) {
      result = xmlhttp.responseText;
      result = result.split(' ')
    }
    if (args == document.cookie){
      orgAlert(args + " " + result[2]);
    }
    else if (args !== undefined) {
      orgAlert(args);
    }
    else{
      orgAlert();
    }
  };
</script>
```

Figure 4.1: How the Flags were Hidden

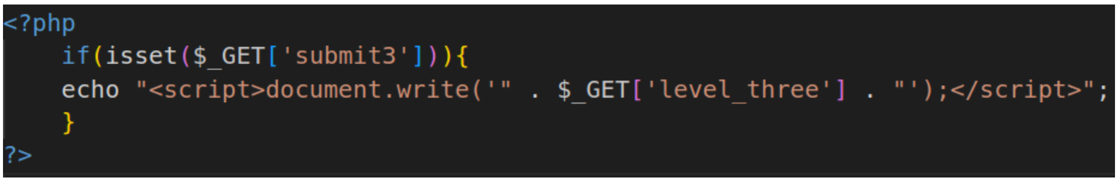
Exercise 1 and 2 - reflected XSS

In the first and second exercises, we wanted to provide the participant with a search bar that reflects the input onto the website. Exercise one and two had different code implementation for this however to provide the users with different challenges, the difference can be seen in Figure 4.2 and Figure 4.3.



```
<div>
  <?php
  if (isset($_GET['submit1'])){
    echo "Sorry ".$_GET['level_one']." was not found.".'<br />';
  }
  ?>
  <form id="form1" action="" method="GET">
    <input type="text" name="level_one" value="">
    <input type="submit" name="submit1" value="Search">
  </form>
</div>
```

Figure 4.2: Input Exercise 1

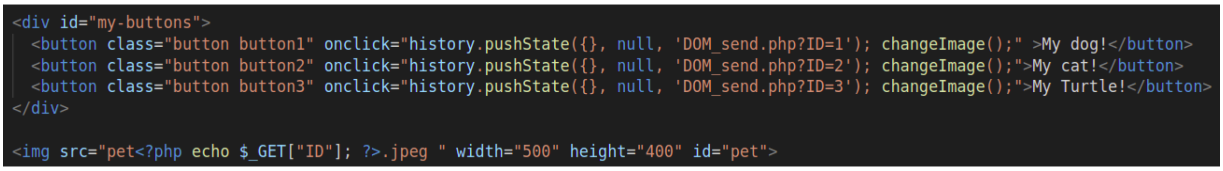


```
<?php
    if(isset($_GET['submit3'])){
    echo "<script>document.write('" . $_GET['level_three'] . "');</script>";
    }
?>
```

Figure 4.3: Input Exercise 2

Exercise 3 - DOM XSS

The third exercise did not provide the participants with an input field, instead, they had to attack the environment variables in the URL. We used the environment variable in the URL to remember the state that the web page was in when the URL was copied, for example, "http://70.34.197.121/XSSDOM.php?ID=1" where the "ID=1" part tells the server that the first image should be presented. To make the site change the image and URL when performing a button click the code shown in Figure 4.4 was used.



```
<div id="my-buttons">
  <button class="button button1" onclick="history.pushState({}, null, 'DOM_send.php?ID=1'); changeImage();" >My dog!</button>
  <button class="button button2" onclick="history.pushState({}, null, 'DOM_send.php?ID=2'); changeImage();">My cat!</button>
  <button class="button button3" onclick="history.pushState({}, null, 'DOM_send.php?ID=3'); changeImage();">My Turtle!</button>
</div>

<img src="pet<?php echo $_GET["ID"]; ?>.jpeg " width="500" height="400" id="pet">
```

Figure 4.4: Button Implementation Exercise 3

Exercise 4 - Mixed XSS

In the final exercise of the CTF, the participant is supposed to steal the cookie from a user of the web page. The purpose of stealing a cookie is to for example hijack a session. In the exercise, the participants are asked to input a malicious URL that he/she has crafted. If the URL is "correctly" formatted and the participant has a web-hook set up correctly, he/she should be able to steal the cookie which is the flag. The web page takes input from the user and then saves the input into a text file on the server. It is then up to the bot to read the file content and visit the URL if valid.

4.3 Hosting

To host the CTFd platform, vulnerable websites, and bot we used VULTR. We chose VULTR because of the low cost, free start credits, and the "one-click" deploy functionality. The setup we did for VULTR was creating an account, receiving our welcome credits, and deploying a server. When the server was up and running we connected to it with Secure Shell (SSH). We configured the server, added a new user to run the bot, and copied the docker images for CTFd and websites along with the actual code for the websites from our local machines to the server using the Secure Copy (SCP) command. The next step was to deploy the docker images which after some minor configuration, port mapping, and path redirection.

4.4 Bot

In the last exercise of the CTF, we wanted to simulate a user visiting the malicious URL that the participants crafted. To simulate a user we created a web bot in Python with the use of the Python module selenium. The abstract functionality of the bot is

1. Make the script run every 5 seconds.
2. If file is not empty read URL from file and erase all content in file.
3. Try to visit the URL using Python requests module.
4. If status code from point 3 is 200, set the cookie to the flag and visit the URL.

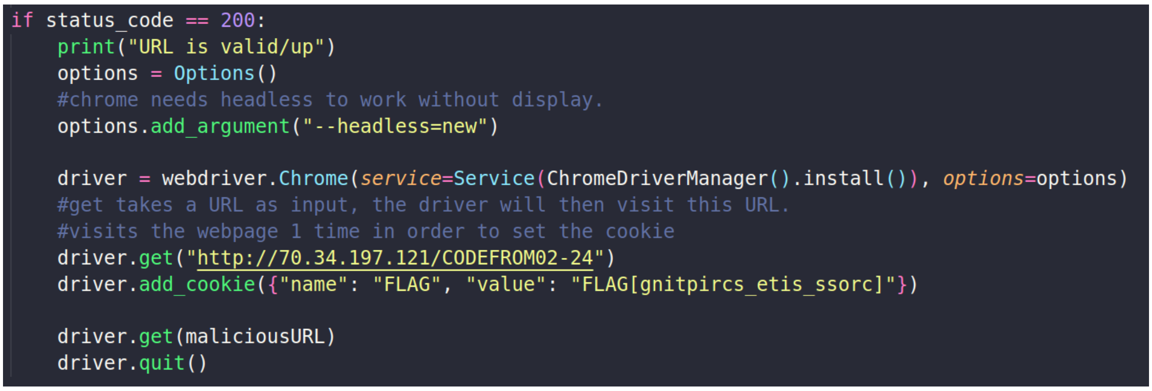
To accomplish the first functionality, the whole script was put inside a while-loop, and a call to Python's sleep function was also added to make the script execute every 5 seconds. This Python script is running on a Linux server that was configured using a shell through SSH. The problem was that every time the SSH session was closed, the process running the script was terminated. This problem was solved by using the Linux screen[2] commands to detach the process from the terminal and enable the process to continue execution after SSH is closed.

The second functionality is simple, the script checks if the file size is equal to zero, if it is the script will sleep for five seconds and then check again. If it is not zero the script will continue to the third functionality.

The Third functionality makes use of the request module. The script makes a request to the URL that has been previously read from the file and saves the status code in a variable called status_code. This code is encapsulated inside a try-except block to prevent the script from crashing if the URL is malformed.

The functionality in the fourth, and last point is where the user simulation occurs.

[2] Screen is a full-screen window manager that multiplexes a physical terminal between several processes[23]



```
if status_code == 200:
    print("URL is valid/up")
    options = Options()
    #chrome needs headless to work without display.
    options.add_argument("--headless=new")

    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
    #get takes a URL as input, the driver will then visit this URL.
    #visits the webpage 1 time in order to set the cookie
    driver.get("http://70.34.197.121/CODEFROM02-24")
    driver.add_cookie({"name": "FLAG", "value": "FLAG[gnitpircs_etis_ssorc]"})

    driver.get(maliciousURL)
    driver.quit()
```

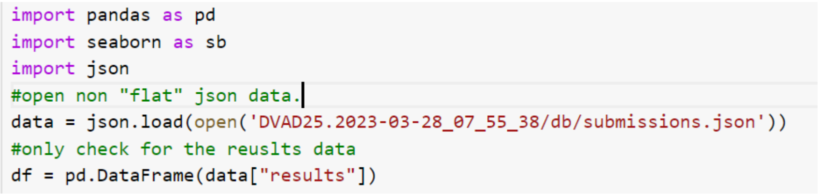
Figure 4.5: Part of the Selenium Python Script

As shown in figure 4.5, it starts with checking the status code previously retrieved. If this status code is 200, we know that the website response was OK. The script then sets the options for the web driver to be headless, this is needed since a web browser cannot run on a machine with no screen connection if this is not enabled. The following line will instantiate the driver, the service variable is set to download the new chrome driver, if there is a new version, every time the script runs. Before visiting the URL the script will visit the main page located at http://70.34.197.121/CODEFROM02-24 to set the cookie to the flag. The script then visits the URL, providing the challenge participant with the cookie if they have their web hook configured correctly in combination with a correct URL.

4.5 Pandas and Seaborn

To present our data/results gathered in the CTF we used pandas and Seaborn. We chose pandas because it provide an easy to use API to handle data. The reason we did not go with other data handling programs such as excel spreadsheets was because we were familiar with pandas and because we wanted to use the Seaborn framework. The reason for working with Seaborn is because of the great compatibility with pandas data frame structure and the easy to use API it provides.

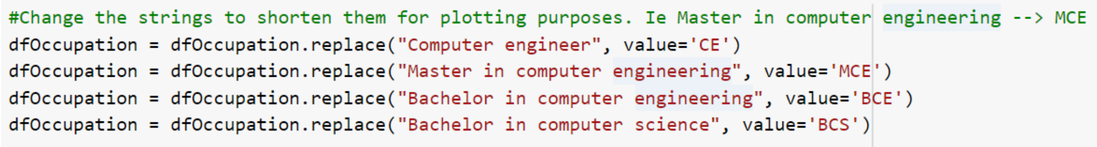
The data was retrieved from CTFd and came in a nested JSON structure, to get the data into a data frame we first opened the JSON file and loaded it into a JSON object called data. Thereafter to get rid of the nesting structure in the data we extracted only the result layer from the JSON object and loaded that data into a panda data frame, this is shown in Figure 4.6.



```
import pandas as pd
import seaborn as sb
import json
#open non "flat" json data.
data = json.load(open('DVAD25.2023-03-28_07_55_38/db/submissions.json'))
#only check for the reuslts data
df = pd.DataFrame(data["results"])
```

Figure 4.6: Loading Data from JSON to Pandas Data Frame

The next step was to process the data. We started with dropping unnecessary columns, such as columns with participant's IP addresses, team, and ID. We then needed to process the data to formats better suitable for Seaborn where answers like "5 (alot)" needed to be changed from the datatype string to integer for example. We did this to enable us to use the data as continuous variables instead of categorical. We also mapped the questionnaire id numbers to the questionnaire titles and replace to long titles with its corresponding acronym to make the data and graphs more readable and later on for the visualization, see example shown in Figure 4.7.



```
#Change the strings to shorten them for plotting purposes. Ie Master in computer engineering --> MCE
dfOccupation = dfOccupation.replace("Computer engineer", value='CE')
dfOccupation = dfOccupation.replace("Master in computer engineering", value='MCE')
dfOccupation = dfOccupation.replace("Bachelor in computer engineering", value='BCE')
dfOccupation = dfOccupation.replace("Bachelor in computer science", value='BCS')
```

Figure 4.7: Replacing Strings with Acronyms

With the data cleaned and structured, we moved on to the presentation part of the data processing. For this we used Seaborn's function count-plot for visualisation.

Chapter 5

Results & Discussion

This Chapter presents the data gathered from the participants, the discussion of said data, and our experiences from the development of the project. In Chapter 3 we presented three requirements along with four factors and justified why these are important for this thesis. With the data gathered, we will discuss, analyze and evaluate how each requirement and factor affected the outcome of the project.

This Chapter has been structured as follows. Section 5.1 presents the recruitment of the test group and the main group. Section 5.2 presents the demography of the test group and the main group. In Section 5.3 we will present the result from the test group's and main group's questionnaires. In the last section 5.4 we will discuss the results from the test group and the main group, only the correlations found will be discussed so some data points will be left out of the discussion.

5.1 Recruitment

In this section, recruitment will be discussed for the test group and the main group.

Test Group

The recruitment of the test group consisted of personal contacts. In total, the test group consisted of three participants, all with a background in IT. One participant is an employee/Ph.D. student at the computer science department of Karlstad Universitet. The two other participants are employees from the IT industry.

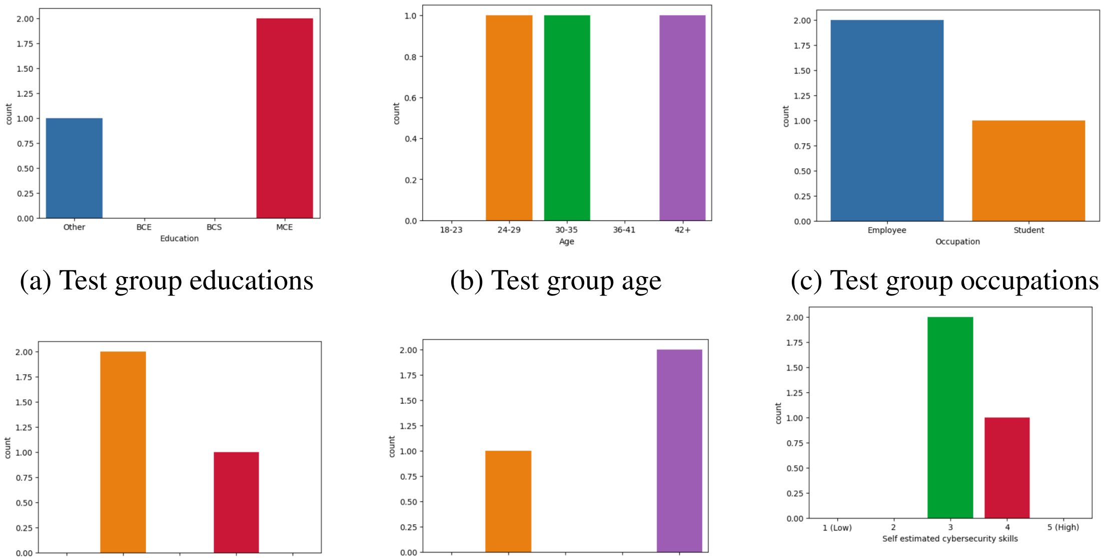
Main Group

The main group consisted of the students in the course DVAD25 and employees from Redpil-Linpro. A notable difference between the participants in the main group was that the students were rewarded with extra points if they finished the assignment which contributed to their final grade in the course.

5.2 Demography

In this section, the demographic data of the participants will be presented for the test group and the main group.

Test Group



(a) Test group educations

(b) Test group age

(c) Test group occupations

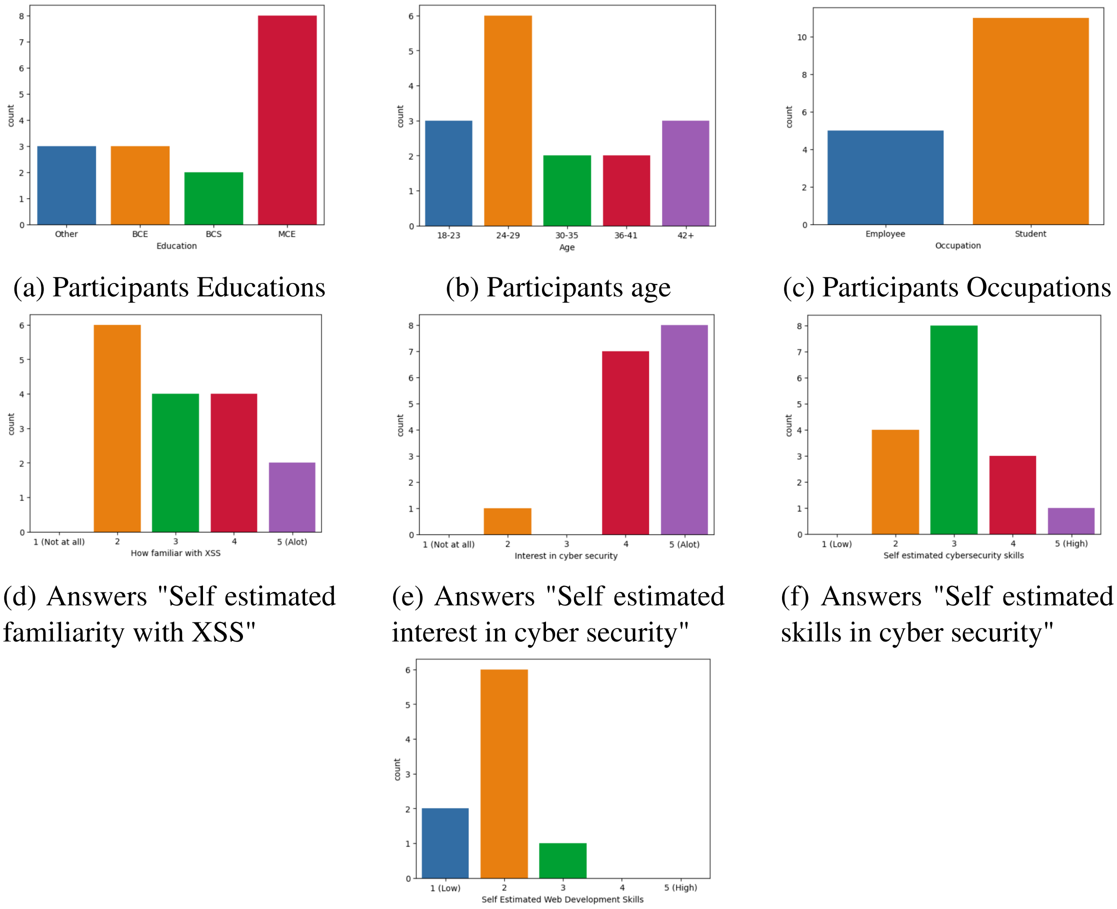
(d) Answers "Self estimated familiarity with XSS"

(e) Answers "self estimated interest in cyber security"

(f) Test group answers "Self estimated skills in cyber security"

Figure 5.1: Demographic Data from the Test Group

Main Group



(a) Participants Educations

(b) Participants age

(c) Participants Occupations

(d) Answers "Self estimated familiarity with XSS"

(e) Answers "Self estimated interest in cyber security"

(f) Answers "Self estimated skills in cyber security"

(g) Answers "Self estimated skills in cyber security"

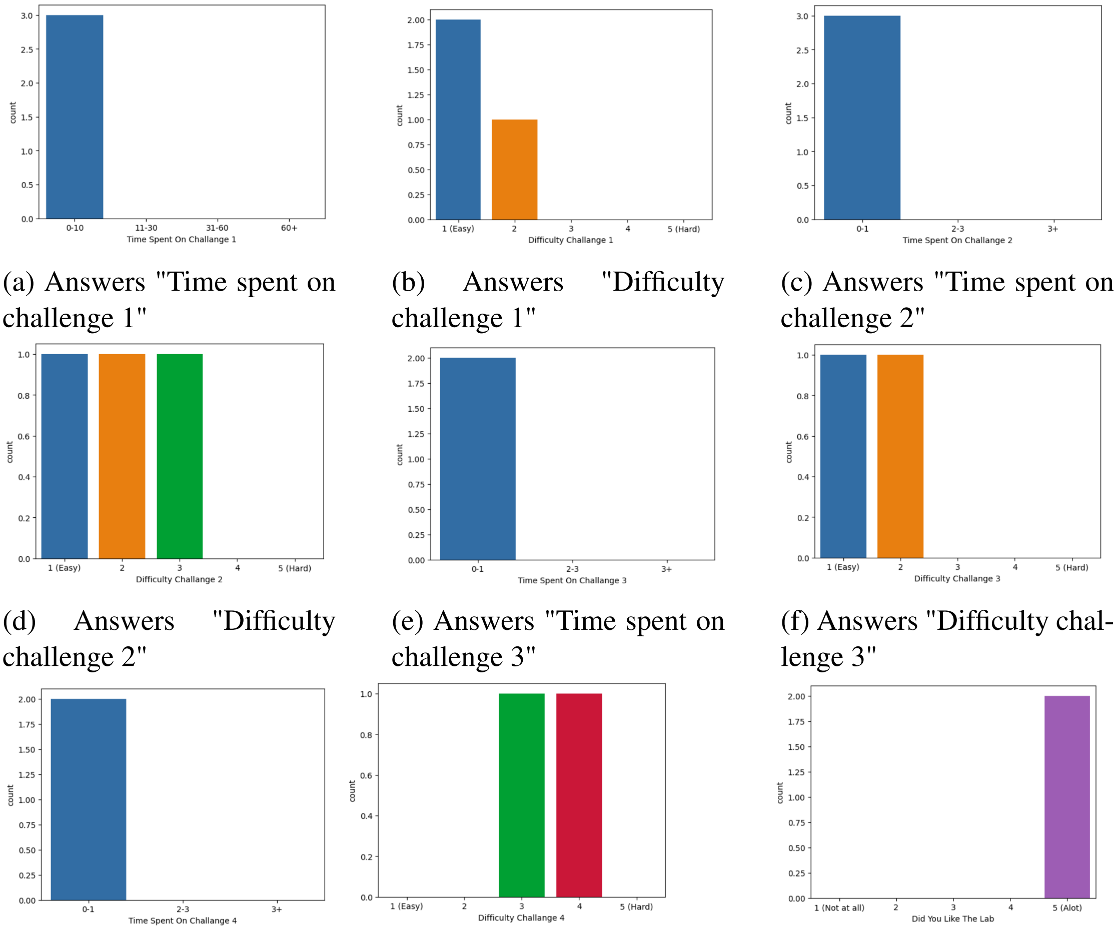
Figure 5.2: Demographic Data from the Main Group

From the results, the following notable outcomes are: The majority of the participants were students in a master's program. The majority of the participants had a high interest in cyber security. The participants had low web development skills.

5.3 Result

This section will present the results from the questionnaires answered by the test group and the main group.

Test Group



(a) Answers "Time spent on challenge 1"

(b) Answers "Difficulty challenge 1"

(c) Answers "Time spent on challenge 2"

(d) Answers "Difficulty challenge 2"

(e) Answers "Time spent on challenge 3"

(f) Answers "Difficulty challenge 3"

(g) Answers "Time spent on challenge 4"

(h) Answers "Difficulty challenge 4"

(i) Answers "Did you like the lab?"

Figure 5.3: Results regarding the test group's challenges

Table 5.1: Answers to questions "Did you cheat?"

"no"
"Fick det inte att fungera, men fick en känsla av att man inte behövde använda en url i sista övningen?"

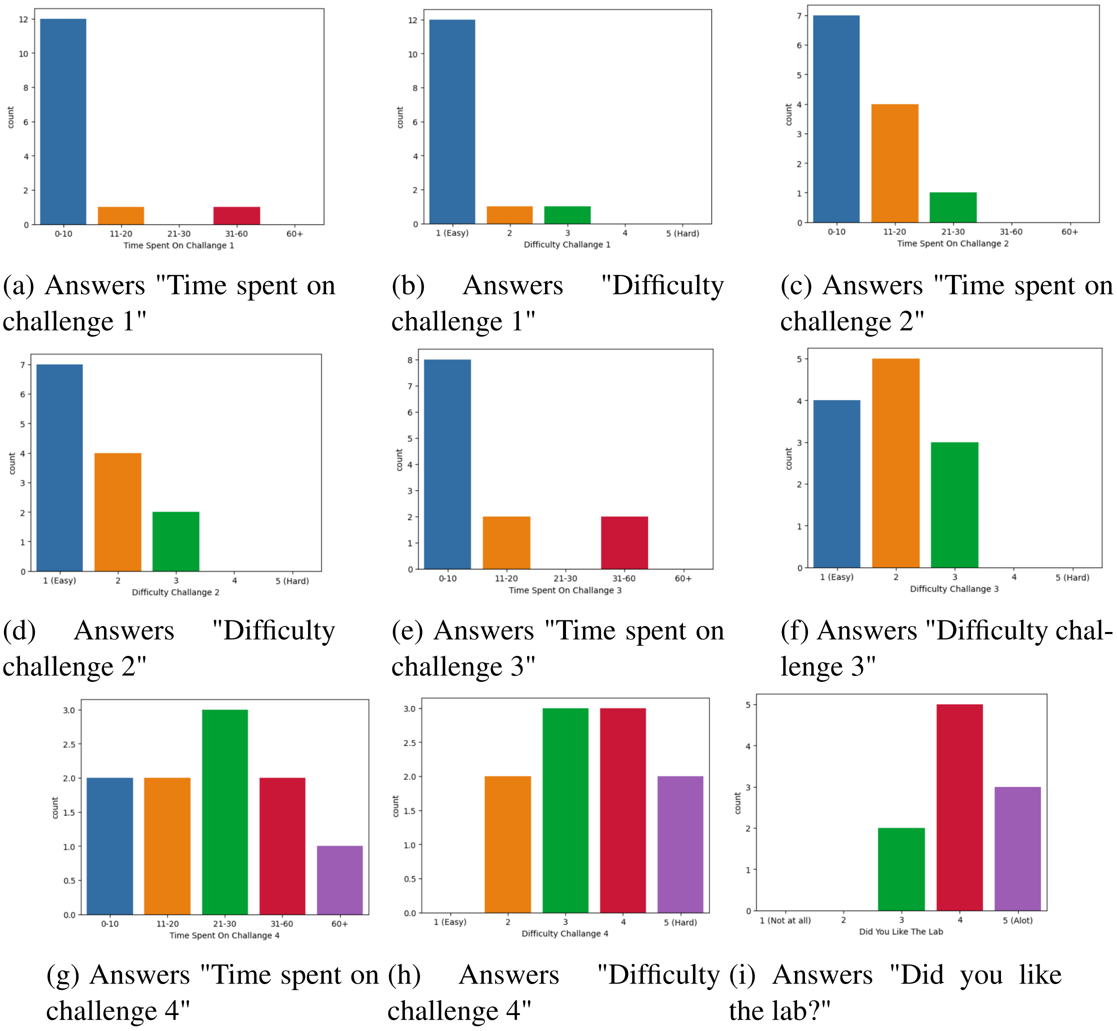
Table 5.2: Answers to questions "What do you think could be improved?"

"not much, looks good"
"Satt med mobilen först (hade ett barn som mest bankar på datorer), och då fick jag inte fram inspectorn. Kanske ska ha med någon kommentar om att det behövs."

Table 5.3: Answers to questions "Other feedback"

"nothing"
"Riktigt kul! Behövde damma av mina web-kunskaper, men det var lagom klurigt. Passade på att försöka googla mig till svaren och kopiera stack overflow, det var inte alls lätt vilket var bra, man ska ju tvingas lära sig något."

Main group



(a) Answers "Time spent on challenge 1"

(b) Answers "Difficulty challenge 1"

(c) Answers "Time spent on challenge 2"

(d) Answers "Difficulty challenge 2"

(e) Answers "Time spent on challenge 3"

(f) Answers "Difficulty challenge 3"

(g) Answers "Time spent on challenge 4"

(h) Answers "Difficulty challenge 4"

(i) Answers "Did you like the lab?"

Figure 5.4: Results regarding the main group's challenges

Table 5.4: Answers to questions "Did you cheat?"

"I searched for how a different site is requsted in javascript. It showed me how I can included it with an img tag. For the flag where you're supposed to change the url I simply went to the console in f12 and asked for the flag instead."
"no"
"no"
"i did not cheat but could have. under the web inspector tool in the browser i found a url: http://70.34.197.121/sdf in the console filed"
"No"
"No, but I had to circumvent my companys outgoing WAF (zscaler) to be able to submit some of the queries"
"No, have not tried"
"No"
"No"

Table 5.5: Answers to questions "Other feedback"

"fun!"
"Free points :D"
Good work guys, hope the exjob goes great!
"Not sure if you know about https://bit.ly/ctf-design but that is a good design document that I often refer to when people ask how to make good CTF challenges"
"Good luck"
"Good work :-)"

Table 5.6: Answers to questions "What do you think could be improved?"

"I think it was a bit too much information. I could more or less already see the solution before clicking the link. Specially for lab 3 which I think was spoiled by the introducory text."
"Set a better password policy, I was able to use a 3 letter password. Make sure all challenges can't be completed with the console. The hint buttons made me nervous as they where so accessible. Might want to up the difficulty slightly, maybe not provide the webhook clue as a free hint."
"It was a bit easy with loads of material available online for help. But a fun beginner challenge for XSS"
"maybe the last challange was difficult because i tried it when tired. but it felt more complex then the others. also i believe leo should give 105 point for this challange even maybe 110. just because of good will :-D"
"This is a very basic XSS challenge, I am not in the target group and not sure what to answer. I guess easy challenges are good for new people."
"The layout, and maybe something to show that the victim actually clicked the link"
"Perhaps a few more tests where you use the bot to show how to steal a cookie from a remote user using phising emails."

Table 5.8: Average Time and Difficulty

Question:	Challenge 1	Challenge 2	Challenge 3	Challenge 4	Total
Time:	8.6	10.2	13.5	21.3	53.7
Difficulty	1.2	1.6	1.9	3.5	2.05

Table 5.7: Questionnaire Averages

Question	Average Answer	Total Submissions
Interest in Cyber Security	4.1	16
Cyber Security Skills	3.2	16
Familiarity XSS	3.2	16
Web Development Skills	2.2	9

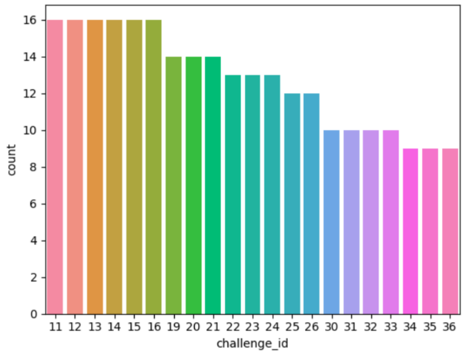


Figure 5.5: Completion of each Challenge in the CTF, where challenge 11 is the first questionnaire

5.4 Discussion

In this section, first, we will discuss the outcome of the test group and explain how their result contributed, we then will discuss the result in regard to the requirements and the most important factors determined in Chapter 3.

Test Group

Questionnaires

A realization from the test group's data was that the questionnaire options regarding time spent on the challenges were too imprecise. All three participants answered 0-1 hours spent on the last three exercises. The fact that their answers were imprecise made it hard for us to use the answers to determine how much time they actually spent. With the data given, we could only know that they spent between 0-3:10 hours in total which made it difficult to decide if we should extend or shorten the length of the exercises. To get better data from the main group we changed these questionnaire options to shorter intervals. The problem concerned the last three exercises and therefore the time interval options for exercises two, three, and four were changed to the same as exercise one, see Figure 5.3a . After reviewing the test group's data we came to the conclusion that it was relevant to know how much prior knowledge the participants had in web development. The reasoning behind why web development experience might be relevant is because the vulnerabilities in the exercises are web-based since they are based on JavaScript injections. Therefore it is relevant to know their prior experience in the web development field. To acquire information about web development skills we added a question regarding participants web development experience to the questionnaire.

Difficulty

When analyzing if the difficulty of the exercises should be increased or decreased, it was observed in the data that we had a broad range of experienced difficulties from the test group. There were two participants who finished the whole CTF, where one reported a difficulty with an average of 2.75, while the second participant reported a difficulty with an average of 1.75. The latter participant reported high prior knowledge in the field, higher than our estimations of prior knowledge for the average participants in the main group. Because of average experienced difficulty was at a reasonable level when taking into consideration the prior experience of the participants of the test group, the conclusion was drawn that the difficulty of the exercises was at a good level.

Main Group

Dropout Rate

Figure 5.5 shows the participation over time, one can observe that the assignment had N = 16 participants where a total of nine participants completed the assignment. Out of these nine participant's a total of seven were students who got extra points in the course DVAD25. A downward trend is shown which was expected and one could argue unavoidable. There is no part of the graph where there is a major increase in dropouts, which could indicate that no specific part of the laboratory assignment was too hard, not likable enough, or too time-consuming.

It is hard to interpret what the total dropout rate signifies. In our laboratory assignment, the total dropout rate was 43,75%, shown in Figure 5.5. If we would have had time to design and construct several laboratory assignments, this data would have been more valuable since we could have compared the dropout rate between the assignments to draw conclusions.

Enough time to implement the exercises

As discussed earlier in the design chapter this project had a strict deadline for the implementation. No project has an infinite amount of time for development, which makes the required time for implementation in regard to the time at our disposal important. The biggest realization when it comes to the factor is that designing laboratory assignments as a set of smaller sub-exercises worked well for this project. This strategy enabled us to scale up or down the assignment which created a flexible way of development in regards to time for development. This strategy enabled us to start with the implementation of the most fundamental exercises, we could with the time left develop more sub-exercises. The ability to scale the project size decreases the risk of missing deadlines. We argue that the use of this technique contributed to us managing to meet the project's deadlines.

Topic not covered by other, preexisting laboratory assignments

As we determined in the design phase that this requirement was met, there will be no further discussion in this chapter.

Success evaluation and flag distribution

When designing the exercises we thought it would be uncomplicated to implement flag distribution in a web environment. It turned out to be more complicated than our original assessment from the design phase. The main problem was the participant's ability to acquire the flags by methods that were not thought of during the design phase. One non-intended method to acquire flags from our original design was through the usage built-in "view page source" command. To solve the "view page source" issue, the flags were hidden in a text file. We decided to stop the development of flag distribution at the point where we believed it was harder to get the flags using non-intended methods than solving the exercise as intended.

Time for the participants to solve the exercises

There are interesting aspects to note regarding time consumption for completing the assignment. For instance, the reported average time consumption for the exercise is shown in Table 5.8

The participants that took the longest time, spent between 133-200+ minutes on the exercises while the student that finished it in the least amount of time spent 0-40 minutes. To analyze what length a laboratory assignment should have for an optimal learning outcome is outside the scope of this report. It is however interesting and important to reflect on if the average time spent seems to be in a reasonable interval. When designing the exercises we aimed for creating an assignment that would take the participants in total between one and three hours. We chose to aim for one to three hours after assessing how much time the other exercises in the course DVAD25 would take to solve since we wanted a similar length on our assignment. Therefore when analyzing the time consumption for our assignment we also reasoned that 53.7 minutes might be too short for the students to get the learning outcome we wanted. After our analysis, we argue that it might be beneficial to add a fifth exercise.

The reason for the assignment being too short for the main group can be traced back to the data from the test group. As mentioned in this section, the test group had too broad intervals in their alternatives for time spent which made it hard to determine how time-consuming the assignment actually was. From the test group's data, we could only conclude that they had spent between 0-4 hours. We could have followed up on these questions in person to get a more accurate answer but that was not thought of at the time before the release of the assignment to the main group.

Relevance for the participants

In the design chapter 3, we argue for XSS being a relevant vulnerability to base an assignment on. Nevertheless, it proves challenging to substantiate our claims with data obtained from the questionnaires. This is due to the lack of questions targeting the factor specifically. To adequately validate our arguments, it would be necessary to include questions regarding the experienced relevance of the assignment. The lack of questions regarding this factor was a shortcoming on our part during the design phase.

How engaging the exercises would be for the participants

How interesting something is, is hard to quantify and what parameters determine is hard to pinpoint, but there are a few things that might be important, for example, time consumption, and difficulty. Since these are covered in other parts of this thesis, they will not be discussed in this section. To enable us to answer and analyze the factor "How Interesting it would be for the participants" we included questions in the questionnaire regarding this factor.

The students have a high interest in cyber security which is shown in Figure 5.8 where it can be seen that the average answer to the question "Interest in cyber security" was 4.048, which we argue will have an influence on their attitude to the assignment.

When the participants were asked the question "Did you like the laboratory assignment?" the average of submitted answers was 4.1. Because of the high average answer to the question "interest in cyber security" it is hard to discern how much the high score on the question "Did you like the laboratory assignment?" is dependent on the quality of the assignment, and how much the answer is dependent on the participant's interest in cyber security. This leads to the question of whether a participant with a low reported interest in cyber security would give us an equally high rating on the question "Did you like the assignment?". To answer this question we would need to find participants with a low interest in cyber security and analyze their data. Nonetheless, we can see this result as a strong positive indication of the quality of the assignment.

Experienced difficulty

As shown in Table 5.8, every exercise experienced average difficulty was higher than the previous exercise. If the first exercise were too difficult some of the participants might have felt discouraged and not continue with the rest of the exercises. The goal was to have increasing difficulty in the exercises since our reasoning was that this would encourage the most amount of participants to complete the assignment. As shown by the result, see Table 5.8 this has been achieved in the participant group we had.

It is hard to say what a perfect average is for the difficulties, but considering the feedback from the participants we drew the conclusion that the difficulty of the exercises was too low. According to the participants, they were given too much information/hints on how to solve the exercises. This feedback was not in line with the result we expected after analyzing the feedback from the test group. The feedback from the test group suggested that the exercises were too difficult, we therefore gave the main group more hints in the exercise descriptions. After the result from the main group, we drew the conclusion that this was the wrong thing to do.

We would argue that difficulty level is the hardest factor to balance. Since we as developers of the exercises already have knowledge about the vulnerability it is hard for us to see the exercises from the participant's perspectives. It might be the first time the participants encounter XSS and therefore it is complicated for us to develop an exercise with difficulty at a good level.

It is interesting to analyze how difficulty relates to how much the participants liked the assignment, and a correlation between the two is shown when analyzing the data. The two participants that submitted the lowest value of three when answering the question on how much they liked the assignment, both answered on all of the challenges that the difficulty was one. Although too small a participation group for a statistical significance for this argument, it is an indication that easy exercises might not be as likable as more challenging exercises. The three participants who answered "Did you like this laboratory assignment?" with a score of 5 had an average difficulty rating of 2,75. This strengthens our argument that a challenging exercise is a better one in terms of likeability.

Furthermore, we can see in the data from the participant's answers to the questionnaires that the participant that reported the highest self-estimated cyber security skills was one of the participants that found the exercises too easy. Even though only one participant gave this answer, it might be an indication that high self-estimated cyber security skills correlates with how difficult they experience the exercises to be, which in turn correlates to how likable the exercise was.

Chapter 6

Conclusion

In this project, the goal was to create a CTF-based laboratory assignment with the purpose of teaching the participants about a specific vulnerability, and for us to analyze requirements and factors that influence the development and the user experience of CTF-based laboratory assignments. In this chapter we will discuss if we managed to achieve our goals, have a brief discussion about the project as whole and relevant topics for future work.

We have created a fully functional laboratory exercise for the course DVAD25, gathered user data from students of the course DVAD25 and Redpill-Linpro employees, analyzed the data, and evaluated the factors along with the requirements. The outcome of the project fulfills the goals of this thesis. All in all the project has been going well and we have not met too many big hurdles. That the project went well is not to say there is nothing to improve. The Future Work explains this topic.

The question we aimed to answer in this thesis was as mentioned in Chapter 1 "How do different factors and requirements influence development and user experience in CTF-based laboratory assignments?". When answering this question we contributed with an analysis of the requirements and factors. The most important results were that a correlation between difficulty and how much the participants liked the lab was found, questionnaire options should not be too broad since that makes the analysis of them less accurate, and distributing flags in web environments is more complex than we first assessed. The question was answered, and therefore the problem of this thesis was solved, as shown in Chapter 5, a detailed discussion on how the requirement and factors influenced user experience and development of a CTF-based laboratory assignment was given.

Future work

Even though the defined goals were fulfilled, it is not to say no more work could be done to improve the laboratory assignment. When it comes to the implementation several improvements could be made, here are the main improvement points thought about during the implementation of the project.

- The flags could be hidden better, making it harder for the participants to use shortcuts.
- An exercise could be added that teaches the participants about stored XSS.
- The graphical interfaces of the exercises could be improved to be more visually appealing.

When it comes to the analysis of the factors there are things that could be done to learn more about the factors and generate more data.

- A database for logs could be set up to gather more information about the participants from their interactions with the exercise websites, for example, GET requests.
- More laboratory assignments could be created to compare the assignments against each other.
- More participants will produce more data for the analysis, which could make the analysis more accurate.

- In hindsight, we can say that expanding the questionnaire with further questions on this topic would have provided useful data to analyze. One example of expansion would be to ask the participants if they thought XSS is a relevant topic for a laboratory assignment.

A big part of any future work in this field would be to validate the result of this thesis. The following list presents our thoughts on ways to validate our results.

- Creating multiple CTF-based assignments of different vulnerabilities and different scores in the factors from the design phase would produce a more comprehensive data set for us to analyze and use for validation.
- Expanding the variety in participants' backgrounds. Such as educational background, interest in cyber security, and web development skills.
- Change the parameters used in this project and observe the new result.
- Use different factors in future reiterations of CTF assignments and observe which factors have the biggest impact on the outcome.

Bibliography

[1] Jemal Abawajy. User preference of cyber security awareness delivery methods. *Behaviour & Information Technology*, 33(3):237–248, 2014.

[2] Danish Jamil and Muhammad Numan Ali Khan. Is ethical hacking ethical? *International Journal of Engineering Science and Technology*, 3(5):3–758, 2011.

[3] Wikipedia contributors. Capture the flag (cybersecurity) — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Capture_the_flag_(cybersecurity)&oldid=1151702164`, 2023. [Online; accessed 8-May-2023].

[4] Lucas McDaniel, Erik Talvi, and Brian Hay. Capture the flag as cyber security introduction. In *2016 49th hawaii international conference on system sciences (hicss)*, pages 5479–5486. IEEE, 2016.

[5] Owasp. Cross site scripting (xss). `https://owasp.org/www-community/attacks/xss/`. Accessed Mars 2023.

[6] Charles Anderson. Docker [software engineering]. *Ieee Software*, 32(3):102–c3, 2015.

[7] Ian Miell and Aidan Sayers. *Docker in practice*. Simon and Schuster, 2019.

[8] Docker. What is a container? https://docs.docker.com/get-started/. Docker Documentation. Accessed April 2023.

[9] Docker. Docker compose overview. https://docs.docker.com/compose/. Docker Documentation. Accessed Mars 2023.

[10] Wikipedia contributors. Yaml — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=YAML&oldid=1148106458, 2023. [Online; accessed 27-April-2023].

[11] Docker. Compose file specification. https://docs.docker.com/compose/compose-file/. Docker Documentation. Accessed Mars 2023.

[12] Google. What is a virtual private server (vps)? https://cloud.google.com/learn/what-is-a-virtual-private-server. Google Documentation. Accessed Mars 2023.

[13] VULTR. The ultimate guide to multicloud strategies with vultr. https://marketing-sales-files.ewr1.vultrobjects.com/ultimate-guide-to-multicloud-strategies.pdf. VULTR Documentation; Accessed Mars 2023.

[14] Selenium. Documentation webdriver. https://www.selenium.dev/documentation/webdriver/. Selenium Documentation. Accessed Mars 2023.

[15] NumPy Developers. Numpy documentation. https://numpy.org/doc/stable/. Numpy Documentation. Accessed April 2023.

[16] NumFOCUS Inc. Package overview. https://pandas.pydata.org/docs/getting_started/overview.html. Pandas Documentation. Accessed April 2023.

[17] The Matplotlib development team. Matplotlib: Visualization with python. `https://matplotlib.org/`. Matplotlib Documentation. Accessed April 2023.

[18] NumFOCUS Inc. Package overview. `https://seaborn.pydata.org/tutorial/introduction.html`. Seaborn Documentation. Accessed April 2023.

[19] Peter Silberman and Richard Johnson. A comparison of buffer overflow prevention implementations and weaknesses. *IDEFENSE, August*, 2004.

[20] Hack The Box. Business ctf 2022: Chaining self xss with cache poisoning - felonious forums. `https://www.hackthebox.com/blog/business-ctf-2022-felonious-forums-write-up`. Hack The Box Challenge Description. Accessed May 2023.

[21] OWASP. Owasp top ten. `https://owasp.org/www-project-top-ten/`. Owasp-top-ten Prject. Accessed April 2023.

[22] Docker. What is docker hub? `https://www.docker.com/products/docker-hub/`. Docker Documentation. Accessed April 2023.

[23] Linux Contributors. screen(1) - linux man page. `https://linux.die.net/man/1/screen`. Linux Man Page. Accessed April 2023.

Appendix

Appendix A

CTFd Docker File

```
...

version: '2'


services:
  ctfd:
    build: .
    user: root
    restart: always
    ports:
      - "8000:8000"
    environment:
      - UPLOAD_FOLDER=/var/uploads
      - DATABASE_URL=mysql+pymysql://ctfd:ctfd@db/ctfd
      - REDIS_URL=redis://cache:6379
      - WORKERS=1
      - LOG_FOLDER=/var/log/CTFd
      - ACCESS_LOG=-
      - ERROR_LOG=-
      - REVERSE_PROXY=true
    volumes:
      - .data/CTFd/logs:/var/log/CTFd
      - .data/CTFd/uploads:/var/uploads
      - .:/opt/CTFd:ro
    depends_on:
      - db
    networks:
        default:
        internal:

  nginx:
    image: nginx:stable
    restart: always
    volumes:
      - ./conf/nginx/http.conf:/etc/nginx/nginx.conf
    ports:
      - 80:80
    depends_on:
      - ctfd

  db:
    image: mariadb:10.4.12
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=ctfd
      - MYSQL_USER=ctfd
      - MYSQL_PASSWORD=ctfd
      - MYSQL_DATABASE=ctfd
    volumes:
      - .data/mysql:/var/lib/mysql
    networks:
        internal:
    # This command is required to set important mariadb defaults
    command: [mysqld, --character-set-server=utf8mb4, --collation-server=utf8mb4_unic

  cache:
    image: redis:4
    restart: always
    volumes:
    - .data/redis:/data
    networks:
        internal:

networks:
    default:
    internal:
        internal: true
```

Appendix B

Consent form

Hi! We are two computer science students, Hugo Andersson and Per Andersson and this capture the flag challenge is part of our bachelor thesis in computer engineering. In this exercise you will learn about and execute cross site scripting attacks on our vulnerable website. We want to collect some data about the before, in betweeen and after the challenge, including some information about you to help us out in the data analysis. All personal identifiers we'll be removed from the data set. Your KAUID will only be used by Leonardo Martucci to assign the bonus, 100 points in the course examination, to you and will then be removed from the data set. The points are awarded after all questions of the challenge are answered. The data you provide will be then encrypted and securely stored. Your questionnaire replies will not affect your score so please answer them truthfully! If you have any questions please send an email us, Hugo and Per, at: questions-on-lab@proton.me You start off with 100 points that you can use to buy hints. The remaining points after completing all the exercises is the amount of points you are rewarded for the laboration. hint 1: Getting started hint 2: Need some help hint 3: Solution By clicking yes, you consent to our usage of your (anonymous data) in our analysis. You may revoke your consent at any time by emailing us at questions-on-lab@proton.me. () YES () NO