John Martin Johnsen

# Automated hint generation for cybersecurity learning

Master’s thesis in Information Security
Supervisor: Basel Katt
Co-supervisor: Muhammed Mudassar Yemin
December 2023

Master’s thesis

**NTNU**
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

John Martin Johnsen

# Automated hint generation for cybersecurity learning

Master’s thesis in Information Security
Supervisor: Basel Katt
Co-supervisor: Muhammed Mudassar Yemin
December 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

# Automated hint generation for cybersecurity learning

John Martin Johnsen

# Abstract

Injection vulnerabilities have long posed a substantial challenge for developers and society at large, particularly as the Internet has become an integral part of our daily lives. These vulnerabilities enable attackers to potentially read, modify, and delete sensitive data, disrupt services, and even execute harmful commands within the exploited application or the operating system. Despite being a long-standing security issue [1], current cybersecurity education primarily focuses on exploiting these vulnerabilities, often with limited or no attention to defence and mitigation. This master thesis seeks to bridge this gap by creating a learning environment which offers guidance on both discovering and remedying injection flaws, and provides learners with valuable support throughout their educational journey.

In this study, an educational artefact with hands-on exercises and automated hint provision was developed to explore the impact of automated hint generation during offensive cybersecurity exercises. By comparing the learner's inject with all known correct solutions and utilising machine learning the artefact generates hints for each individual. Analysing user log data, including injects, time spent, and objectives completed, between groups with and without hints, to assess the effectiveness of the artefact. As well as assessing the learners' knowledge through surveys, to further evaluate the effect of the artefact. Results from questionnaires are used to assess perceived learning experience and identify areas of improvement. All this is done in order to comprehensively evaluate the artefact.

While the generated hints had no significant impact on learning outcomes or provided any benefits, all participants showed improvement from pre-survey to post-survey, suggesting an overall impact of the learning environment. While results showed no significant differences, participants in the control group expressed the belief that hints could be beneficial. The results are overall inconclusive because of technical issues during testing resulted in limited data, and uncertainties in regards to the data's validity and reliability. Therefore, future works needs to optimise hint generation, test anew, and reevaluate the artefact on a larger scale to draw conclusive findings.

# Sammendrag

Injeksjonssårbarheter har lenge utgjort en betydelig utfordring for utviklere og samfunnet generelt, spesielt siden Internett har blitt en integrert del av hverdagen vår. Disse sårbarhetene gir angripere muligheten til potensielt lese, endre og slette sensitiv data, forstyrre tjenester og til og med eksekvere skadelige kommandoer i den utnyttede applikasjonen eller operativsystemet. Til tross for å være et langvarig sikkerhetsproblem [1], fokuserer dagens opplæring innen cybersikkerhet primært på utnyttelse av disse sårbarhetene, ofte med begrenset eller ingen tanke på forsvars- og motvirkningstiltak. Denne masteroppgaven søker å tette dette gapet ved å skape et læringsmiljø som tilbyr veiledning i både å oppdage og fikse injeksjonsfeil, og gi elevene verdifull støtte gjennom hele læringen.

I denne studien ble det utviklet et pedagogisk artefakt med praktiske oppgaver og automatiske hintgivning for å undersøke effekten av automatisk hintgenerering under offensive cybersikkerhetsoppgaver. Ved å sammenligne elevens injeksjon med alle kjente riktige løsninger og bruk av maskinlæring genererer artefaktet hint til hver enkelt. Analyse av brukerloggdata, inkludert injeksjoner, tid brukt og fullførte oppgaver, mellom grupper med og uten hint, for å vurdere effektiviteten til artefaktet. I tillegg til kartlegging av kunnskapen til elevene før og etter gjennom undersøkelser for å ytterligere vurdere effekten til artefaktet. Resultater fra spørreskjemaer er brukt for å vurdere elevenes læringsopplevelse og identifisere forbedringsområder for artefaktet, for å helhetlig evaluere artefaktet.

Selv om de genererte hintene ikke hadde noen signifikant innvirkning på læringsutbytte eller ga noen fordeler, viste alle deltakerne forbedring fra forhåndsundersøkelsen til etterundersøkelsen, noe som antyder en påvirkning som følge av læringsmiljøet. Mens resultatene ikke viste noen signifikante forskjeller, uttrykte deltakerne fra kontrollgruppen troen på at hint kunne ha vært nyttig. Resultatene er generelt uklare på grunn av tekniske problemer under testingen som førte til begrenset data og usikkerhet til dataens gyldighet og pålitelighet. Derfor må fremtidig arbeid optimalisere generering av hint, teste på nytt og reevaluere artefaktet på en større skala for å trekke avgjørende konklusjoner.

# Preface

This thesis is the final delivery of a master's degree in Information Security at the Norwegian University of Science and Technology (NTNU) in the faculty of Information Technology and Electrical Engineering. The work spanned from January to mid-December 2023 and is a contribution to the field of hint generation within cybersecurity, specifically addressing the challenge of providing tailored hint for each individual learner in the process of learning injection vulnerabilities. The chosen topic focuses on applying hints during hands-on learning in an offensive security lab environment, leveraging machine learning to generate customized hints.

John Martin Johnsen

Lillehammer, Friday 15th December 2023

# Acknowledgements

I would like to express my sincere gratitude to my supervisors, Basal Katt and Muhammed Mudassar Yemin, for their invaluable insights, guidance, and tips throughout the development of this thesis. Their expertise and support greatly enriched both the content and writing process.

A special thanks to Kirsi Helkala at the Norwegian Defence Cyber Academy and Christian Johansen and Shao-Fang Wen at NTNU for generously allowing me to conduct tests on their students and for their contributions to the structure of the developed system.

I extend my appreciation of my friends Ole Joachim and Torkel, whose technical expertise and insights were instrumental in shaping this work. Additionally, I want to thank my employer, Advania, and colleagues for their support, which has enabled me to dedicate time and effort to completing this thesis.

John Martin Johnsen
Lillehammer, Friday 15th December 2023

# Contents

# Figures

# Tables

# Code Listings

# Acronyms

**AI** Artificial Intelligence. 3, 23, 55

**API** Application Programming Interface. 36

**CISA** Cybersecurity and Infrastructure Security Agency. 1

**CTF** Capture The Flag. xv, 7–11, 15, 20, 36, 39, 50, 51

**CVSS** Common Vulnerability Scoring System. 1

**DVWA** Damn Vulnerable Web Application. 12, 13

**GPT** Generative Pre-trained Transformer. 19, 21, 23, 24, 35, 36, 55

**LLM** Large Language Model. 19, 21, 23, 24

**MDP** Markov Decision Process. 15

**NLP** Natural Language Processing. 23

**NTNU** Norwegian University of Science and Technology. vii, ix, xv, 37, 39, 42, 44–46, 48–55, 57

**OWASP** Open Web Application Security Project. 1, 7, 12

**RLHF** Reinforcement Learning from Human Preferences. 24

**SANS** SysAdmin, Audit, Network, and Security. 1, 12

**SQL** Structured Query Language. 14, 16

**SSRF** Server-Side Request Forgery. 12

**SSTI** Server-Side Template Injection. 12, 42, 49, 55

**VPN** Virtual Private Network. 12

**XVWA** Xtreme Vulnerable Web Application. 12, 13

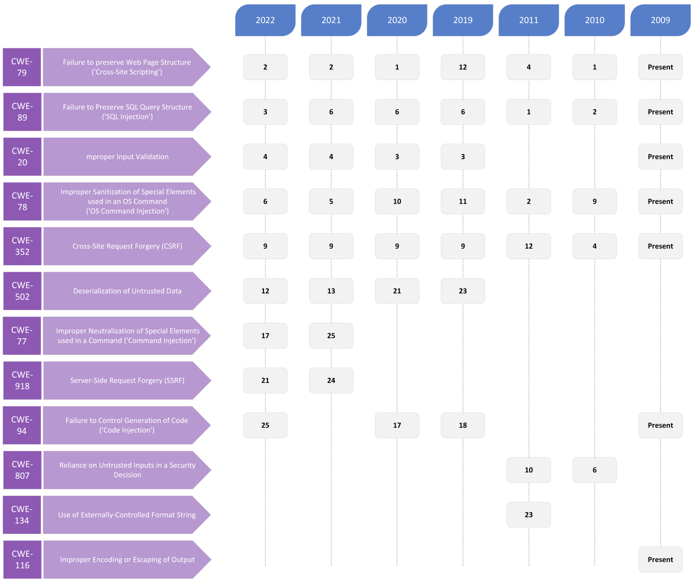
# Chapter 1

# Introduction

## 1.1 Topic Covered by the Project

Injection vulnerabilities have long posed a substantial challenge for developers and society at large, particularly as the Internet has become an integral part of our daily lives. The significance is emphasised by its persistent presence in the Open Web Application Security Project (OWASP) Top 10 Web Application Security Risks, where injection vulnerabilities have consistently ranked within the top three since the inception of the list [1].

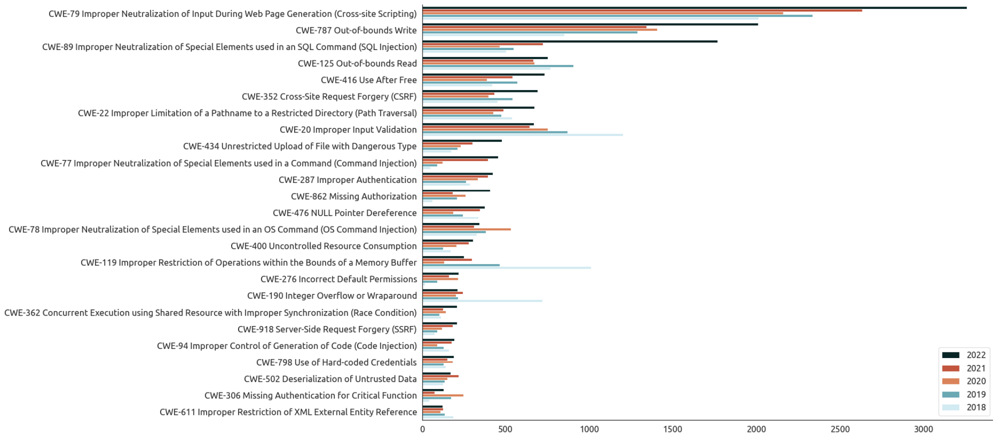
In 2009, MITRE in collaboration with SysAdmin, Audit, Network, and Security (SANS) institute created a compiled list of the most common and easily exploited vulnerabilities, and grouped similar vulnerabilities together [2]. Subsequently, they weighted the Common Vulnerability Scoring System (CVSS) score, and vulnerabilities in Cybersecurity and Infrastructure Security Agency (CISA)'s list of actively known exploited vulnerabilities into the ranking [3].



| | | 2022 | 2021 | 2020 | 2019 | 2011 | 2010 | 2009 |
|---|---|---|---|---|---|---|---|---|
| CWE-79 | Failure to preserve Web Page Structure ('Cross-Site Scripting') | 2 | 2 | 1 | 12 | 4 | 1 | Present |
| CWE-89 | Failure to Preserve SQL Query Structure ('SQL Injection') | 3 | 6 | 6 | 6 | 1 | 2 | Present |
| CWE-20 | mproper Input Validation | 4 | 4 | 3 | 3 | | | Present |
| CWE-78 | Improper Sanitization of Special Elements used in an OS Command ('OS Command Injection') | 6 | 5 | 10 | 11 | 2 | 9 | Present |
| CWE-352 | Cross-Site Request Forgery (CSRF) | 9 | 9 | 9 | 9 | 12 | 4 | Present |
| CWE-502 | Deserialization of Untrusted Data | 12 | 13 | 21 | 23 | | | |
| CWE-77 | Improper Neutralization of Special Elements used in a Command ('Command Injection') | 17 | 25 | | | | | |
| CWE-918 | Server-Side Request Forgery (SSRF) | 21 | 24 | | | | | |
| CWE-94 | Failure to Control Generation of Code ('Code Injection') | 25 | | 17 | 18 | | | Present |
| CWE-807 | Reliance on Untrusted Inputs in a Security Decision | | | | | 10 | 6 | |
| CWE-134 | Use of Externally-Controlled Format String | | | | | 23 | | |
| CWE-116 | Improper Encoding or Escaping of Output | | | | | | | Present |

**Figure 1.1:** Injection trend

Figure 1.1 illustrates the evolving landscape of injection vulnerabilities, revealing a growing number of diverse injection types. The Stack conducted an analysis of the top 25 vulnerability groups between 2018 and 2022, examining their annual vulnerability count, as depicted in Figure 1.2. This data underscores the escalating importance of addressing injection vulnerabilities and the pressing need for effective mitigation strategies. Furthermore, it highlights the critical role of educating developers and cybersecurity personnel in understanding, preventing, and mitigating these vulnerabilities.



Graphic: Nick Sexton for The Stack. Source: nvd.nvist.gov

**Figure 1.2:** The Stack's analysis of vulnerabilities

For those who are unfamiliar with these types of vulnerabilities, acquiring knowledge about how to discover and mitigate them can be a significant challenge. Understanding how injection vulnerabilities manifest in web applications and how to effectively mitigate them is essential for their eradication. With the increasing use of Artificial Intelligence (AI) in code generation, it's become evident that AI systems do not always prioritise security [4]. However, if asked if the code generated is secure, it can recognise the code as insecure. Still, it can be directed to implement specific security measures if explicitly instructed [5]. A sandbox provides a controlled environment where users, whether they are students or developers, can gain the knowledge to identify and mitigate these vulnerabilities while allowing them to test potential attacks.

This thesis will focus on developing an injection lab environment, a sandbox, which covers steps in the exploitation process and security mitigations for injection vulnerabilities. The overall goal is to assist the learning process by providing tailored and automatically generated hints during the learning process.

## 1.2 Problem Description

> *Injection vulnerabilities persist as a significant threat, necessitating comprehensive understanding and effective mitigation strategies, especially in light of existing educational gaps and a shortage of resources tailored for beginners, and the emerging need for automated feedback to enhance offensive cybersecurity learning.*

Injection vulnerabilities pose a significant threat, enabling attackers to potentially read, modify, and delete sensitive data, disrupt services, and even execute malicious commands within the exploited application or the operating system. These threats persist as long as applications accept user-controlled data, making it an ongoing concern. Despite being a long-standing security issue [1], current programming and cybersecurity education primarily focuses on exploiting these vulnerabilities, often with limited or no attention to defence and mitigation.

Understanding how injection vulnerabilities occur, and effective mitigation strategies is paramount. Securing user-controlled data is a complex task, requiring not only knowledge of proper sanitization and security mechanisms [6], but also the ability to anticipate and address potential attack vectors. Without effective and correct sanitization and security measures, malicious input data can alter an application's expected behaviour, manipulate stored data, and, in the worst-case scenario, take full control over the application and its underlying system. This type of knowledge can be used unethically to gain unauthorised access to systems and other malicious purposes. That being said, knowledge itself is not inherently good or bad, it is the intention of the person utilising the knowledge which is.

For individuals new to the field of cybersecurity and aspiring developers, gaining structured, beginner-level knowledge of injection vulnerabilities can be a challenging endeavour, often necessitating payment for services like Pentesterlab [7] and Pentester Academy [8]. Moreover, while the numerous free online resources allow users to explore similar vulnerabilities [9–13], there is a noticeable scarcity of resources that delve deep into a specific category of vulnerabilities.

Since providing personalised feedback to each individual student is a time-consuming task for instructors and does not scale well with large groups [14], recent research have explored the possibility of automating feedback [14–23]. In the field of cybersecurity education, the focus has primarily been on programming feedback [20–23], with emerging research on generating feedback for post-training within offensive cybersecurity [14, 16]. To the best of my knowledge, there has been no published research on providing automated feedback during offensive cybersecurity learning. This thesis seeks to bridge this gap by offering comprehensive guidance on both discovering and remedying injection flaws, and providing learners with valuable support throughout their educational journey.

## 1.3 Justification, Personal Motivation, and Benefits

The primary objective of this project is to simplify the learning process for novice cybersecurity professionals, enabling them to gain a comprehensive understanding of injection vulnerabilities and how to effectively mitigate them. In the ever-evolving landscape of web application security, injection vulnerabilities remain a persistent concern [24, 25]. While this thesis does not directly aid in the shortage of cybersecurity personnel [26] it does aid in the aspect of helping beginners to understand one type of injection flaws, possibly helping them broaden their knowledge. As Sun Tzu wisely noted:

> *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.* [27]

This ancient wisdom highlights the critical importance of knowing the adversaries and the strengths and weaknesses of your own defences. By empowering future cybersecurity professionals with knowledge of injection vulnerabilities, they are enabled to comprehend where and how these threats may surface, where and what to monitor, and how to implement effective countermeasures.

The security of an application, which prevents unauthorised data exfiltration, manipulation or deletion, safeguards the confidentiality, integrity, and availability of the application. As a side-effect also upholding the owner's reputation. Therefore, enhancing knowledge in this domain is not just beneficial; it's a necessity.

This project is not only a response to the ongoing issue of injection vulnerabilities, but also a commitment to address the educational gaps in the field of cybersecurity. By providing a structured and effective learning environment, along with the generation of helpful hints, it aims to equip learners with the knowledge to understand, identify, and mitigate injection vulnerabilities. This educational journey will focus specifically on the sub-category of template injection flaws, shedding light on a niche yet impactful aspect of injection vulnerabilities.

By offering a hands-on learning experience, the project aims to bridge the knowledge gap for emerging cybersecurity professionals. This approach makes it easier for individuals to identify and mitigate vulnerabilities in both development and production environments, ultimately contributing to the creation of more secure applications.

## 1.4 Research Questions

Derived from the problem description the main research question that the thesis will try to answer is:

> *What benefits does providing feedback in the form of generated hints during the injection exploitation process, in the context of learning aid beginners in learning its concepts?*

To answer the research question the following subsidiary questions have been derived to assist answering:

- **RQ1** Did the hints contribute to less time and attempts spent on the challenges?
- **RQ2** How useful is the feedback in improving students' knowledge?
- **RQ3** How can the feedback be improved?

## 1.5 Contributions and structure of this thesis

The primary stakeholders are cybersecurity researchers, analysts and developers, all that can contribute to cybersecurity. Although there are some research done on using machine learning to assist cybersecurity education it has mainly been focused on its application in programming education. Providing hints during lab exercises have proven successful in earlier projects [17] in assisting the learning and comprehension process. Application of dynamic guidance could help cybersecurity personnel in quicker understanding why injection vulnerabilities appear and how to mitigate them, which is still a prevalent issue [25]. Other interested parties in this study could also include educational researchers which aim to apply automated data-driven feedback during exercises. Through testing on cybersecurity students this thesis shows that data-driven hints generated by a machine learning model have the potential to enhance the learning process within offensive cybersecurity and reduce the toll of instructors aiding each individual student. There is still room for improvement of the hints and understanding their effects on different levels of competence.

The rest of the thesis is structured as follows: Chapter 2 describes the background for this thesis, as well as state of the art within cybersecurity education and digital educational platforms and scaffolding. The chapter ends with an introduction to the chosen algorithms and technology for this project. Chapter 3 presents the chosen methodology for answering research questions of the thesis, how data was collected and how it was analysed. To evaluate the proposed system, it was deployed in three test cases, where one only included the final challenge. Chapter 4 thoroughly explains the design and implementation of the proposed learning environment, detailing its key components, features, and workflow. Chapter 5 examines the results and feedback, assessing how hints influenced these outcomes. Findings and their significance in addressing the research questions are highlighted. This chapter also addresses limitations of the research and factors that might have influenced the results. Chapter 6 concludes this thesis and proposes possible future works.

# Chapter 2

# Background and literature

## 2.1 Background

The field of cybersecurity is intricately connected to offensive and defensive security, as well as programming. Proficiently teaching these subjects is vital for understanding the elimination of security flaws. To tackle this, we must comprehend the origins, common location, and potential exploitation strategies for these vulnerabilities. OWASP has been at the forefront of cybersecurity, and has maintained a list of top cybersecurity problems and vulnerability types since 2003. Their list takes into account both the impact and the frequency of these issues, with injection flaws consistently ranking among the top positions since 2007 [1].

Since the beginning of research in cybersecurity education, online learning environments, and more recently, machine learning, have spurred innovation in this field. These approaches can be broadly divided into three categories: locally installed learning environments, online platforms, and a hybrid of the two.

Firstly, there are locally installed learning environments. These depends on the learner to be able to set up the learning environment themselves locally on their own computer. An example of such a platform could be OWASP's Webgoat [12], which gives an introduction to finding and exploiting common web application vulnerabilities. Many such environments are available in docker containers or as virtual machines. These will be discussed in more detail later.

Secondly, there are learning environments that are always available online. These environments are often gamified in the form of CTF where the learner is tasked to acquire a flag (a string or hash of a string) by using or misusing the functionality of the system. While other online learning environments are structured as a course with small assignments throughout.

The third category, a hybrid of the two aforementioned categories. This category has become more used as the availability of cyber ranges has increased, and the need for computational power and storage for machine learning has increased. Such an approach is favoured because it enables the instructor to be able to control who has access to the system, limits the strain on the system and makes it easier to observe usage of the system. Mirkovic et al. [28] deployed their system, ACSLE, on the cyber ranges EDURange and Deterlab, and used it to collect and monitor students progress and commands in hands-on cybersecurity exercises, which enabled instructors to intervene when students were struggling.

Research on automated feedback, not strictly focused on using automated data-driven feedback will be mentioned in section 2.2.2, as they implicitly contribute to and give a more correct picture of the state of the art in the field of educational scaffolding.

The following section delves into the specific categories of learning environments, shedding light on their characteristics, advantages, and limitations. This exploration sets the stage for understanding the importance of feedback and hints in enhancing the learning process.

## 2.2 Related work

This section explores the popular trend of learning through CTF competitions and gamification of learning, delving into their impact on the learning process. These concepts heavily influenced the design of the hands-on exercises in this thesis. Additionally, we will delve into popular learning environments, examining their content and the support features they offer. The section also explores research on educational feedback systems and their effects, providing an overview of available resources, the current state of educational feedback systems, and the diverse fields in which these systems have been tested.

In recent years, one of the popular methods for learning cybersecurity has been through CTF challenges, which adopt a practical approach rather than the traditional theory-first approach. These competitions challenge the participants to exploit different vulnerabilities to circumvent intended usage of an application and its security mechanisms. The challenge involves discovering a specific string in a given format (flag) and submitting it as proof of solving the challenge. These challenges often reward points; the more challenges a participant completes, the more points they accumulates. The points are then used to rank challengers against each other, making a competition out of it. Some competitions also allows the challengers to spend points to unlock hints to aid them in completing challenges.

In 2012, Fulton et al. [29] stated that only a handful of CTF competitions have identified specific educational outcomes or goals. The absence of these goals can hinder participants' ability to prepare effectively, measure their success in understanding the concepts presented in the challenges and in acquiring the relevant skills.

Gamifying the learning of technical concepts in cybersecurity has shown some positive effects [30–33]. A popular way of such gamifying is through CTF competitions. CTF challenges can be divided into two groups: jeopardy and attack-and-defend. The former involves presenting several challenges to the participants, while the latter requires participants to both defend a system or network and attack the same system controlled by opposing teams. The attack-and-defend style CTF requires a lot more knowledge about cybersecurity vulnerabilities and how to mitigate them to be successful. Therefore, is this style of competition not well suited for beginners. Research by Karagiannis and Magkos [34] highlights the significance of students' perceived learning as an important factor for the successful implementation of CTF challenges in an educational context. High perceived learning also appears to correlate with high information retention rates. They structured the learning material as a linear progression, gradually challenging students to apply their acquired knowledge. Their findings indicate a strong correlation between perceived learning, attention to the subject matter, and satisfaction in acquiring new skills. To ensure effectiveness across all participant levels, it's essential that the provided learning material is comprehensive, allowing all students to understand the concepts and enable them to practise the learned skills. Additionally, relating exploits and vulnerabilities to real-world incidents can amplify the perceived importance of the topic and potentially boost students' motivation.

One drawback of using CTF to teach students about cybersecurity concepts is that they rarely provide information about the concepts and security flaws prior to the challenge, or afterwards. McDaniel et al. [30] suggested that basic concepts could be addressed through a series of introductory challenges where explanations of the problem are provided. This approach could also include guidance on correctly securing and mitigating the vulnerabilities. Another limitation of CTF competitions is the infrequent availability of hints on challenges for when participants are in need of assistance. In the cases where hints are provided, the usage results in a small reduction in their total points. This penalty is likely implemented to discourage participants from excessively relying on the hint system and to encourage them to conduct independent research on potential vulnerabilities. Hints are most likely intended for participants who are completely stuck and need assistance to make progress.

However, turning the learning process into a competition encourages and motivates participants to attempt more difficult challenges, thereby enhancing their knowledge and performance [32]. The authors emphasised that in an educational context, the focus of CTF should be on guiding students in learning rather than measuring skill levels. The authors also suggested having a lesson-learned discussion after the competition to encourage the students and teams to share knowledge. Therefore, the sequence of challenges within the same topic should gradually become more challenging to enforce a deeper understanding of the covered concepts while still maintaining the motivational gamification of the learning. These considerations are crucial to prevent the development of a "CTF mindset", where the primary focus is on completing as many challenges as possible within the limited competition time, often resulting in the reliance on automated tools, which undermines the learning aspects.

As mentioned in [33], creating a connection between game mechanics and high-level educational objectives can be challenging. However, with CTF challenges, it is possible to design tasks directly linked to learning objectives. There is no need to create an entire game around these objectives; instead, technical tasks can be designed to help students understand various aspects of cybersecurity.

Several other studies have shown that gamification has a positive impact on learning, resulting in increased motivation, engagement, and performance [35–38]. For instance, research conducted by Morales-Trujillo et al. [38] revealed that students found their learning experience more engaging and satisfying when gamification was introduced, including points and leaderboard, compared to a group that completed the same task without these features. Gamification also led to increased social interaction among students and their immersion and enjoyment. While some students reported increased stress and anxiety due to gamification, these negative effects were considered outweighed by the benefits it offered.

In response to the limitations mentioned in [29], the authors of [39] employed various interventions to guide and assist students in achieving their learning objectives during a hackathon within their online course. Lecture intervention provided the students the basic knowledge required for the challenge. The research findings indicated a strong correlation between the effectiveness of this intervention and its perceived relevance and comprehensibility. The second intervention introduced was feedback from an expert, which improved teamwork and resolved misunderstandings in the team regarding the learning material and tasks, thus increasing team effectiveness. This form of progressive feedback enhanced the teams' ability to handle new tasks due to its timely delivery during the hackathon. The last type of intervention was a team management plan designed to support teams in organising their cooperation, setting task deadlines, and assigning responsibilities and tasks among team members, further enhancing effectiveness.

Motivated by the motivational and educational facets of CTF competitions, this thesis aims to leverage gamification principles to enhance the learning experience.

Following the recommendation of [32], the focus is on guiding students in learning rather than solely measuring skill levels, and to counter "CTF mindset" a point system will not be implemented. By structuring and presenting the learning material as a linear progression, the learning environment can cater to participants with varying skill levels, making the it accommodating for a broader target group. Gradually challenging students to apply their acquired knowledge can enhance the sense of mastery, leading to increased perceived learning and higher information retention rates. As suggested by McDaniel et al. [30], the thesis will take the students through the basics of template injection, as well as providing guidance on correctly securing and mitigating these vulnerabilities.

Furthermore, inspired by the suggestions of Karagiannis and Magkos [34], the learning environment allows students to tackle challenges in an environment resembling a real-life website. This emulation of real-world scenarios aims to underscore the relevance of the subject, potentially increasing students' motivation.

In alignment with the insight from [39], the thesis deploys interventions by providing hints during the students' engagement. Additionally, to encourage the usage of hints, they will be provided automatically without penalising the student. Rewarding the students points and scoring them will not be implemented to maintain a high-quality learning environment, as the number of challenges required for scoring might reduce the quality of each challenge and the overall quality of the learning environment.

The following section will offer an insightful overview of the current state of cybersecurity education and digital learning environments. We will also explore key contributions from other research in the realm of digital learning, providing a comprehensive perspective of this dynamic field.

### 2.2.1 Learning environments

#### Virtual machines

Locally installed learning environments for cybersecurity issues are labs that are meant to be downloaded and run on one's own computer. Learners can progress through the challenges at their own pace, with the progress being saved in a local database, allowing them to resume their work whenever they wish. Over the last decade, several hands-on local learning labs for cybersecurity have been published, such as [10–13]. We will delve into these in more detail in the following subsection.

Some labs do provide users with challenges, accompanied by a small amount of learning material and other scaffolding [10, 12, 13], while others offer challenges that the users must tackle on their own [9, 11]. This distinction suggests that some labs are tailored to more experienced users, while others aim to guide and assist users in understanding vulnerabilities. All of the mentioned labs offer challenges covering various vulnerability categories, and therefore not being very detailed in explaining each security issue.

Xtreme Vulnerable Web Application (XVWA) [11] is an intentionally vulnerable web application written in PHP. It provides a brief explanation of each vulnerability, and covers several common web application weaknesses along with a few modern attack techniques. Additionally, it includes non-traditional vulnerabilities such as Server-Side Template Injection (SSTI) and Server-Side Request Forgery (SSRF). XVWA [11] does not provide hints or direct assistance to learners, but it offers a link per challenge, explaining the vulnerability.

A similar project Damn Vulnerable Web Application (DVWA) [10] addresses some of the most common web vulnerabilities. DVWA allows users to adjust the security level to change the exploitation difficulty level. It offers links to resources related to the vulnerabilities and provides the option to display one hint for each challenge per difficulty level. Furthermore, DVWA allows learners to view the source code that needs to be exploited.

OWASP have created two local labs, namely multillidae-II [13] and WebGoat [12], with a primary focus on the OWASP Top 10. Multillidae-II incorporates vulnerabilities from SANS Top 25 Programming Errors, addressing the most critical cybersecurity issues. In contrast, WebGoat covers vulnerabilities from all the OWASP Top 10 lists and offers more detailed explanations about each vulnerability. It also tracks students' progress through each vulnerability category. Multillidae-II, on the other hand, provides brief descriptions of the vulnerabilities, it offers three different security levels for students to bypass, along with one hint per difficulty level per challenge. One of WebGoat's strengths, when compared to multillidae-II, is that it requires the students to submit a proof of completion, whereas multillidae-II relies on multiple-choice assessments. Furthermore, WebGoat has been made available in an online environment as a room on TryHackMe, removing all setup required by the learner.

Wang et al. [40] developed 12 labs, which they consolidated into their IT-SEED project. They meant that the learning advantages of hands-on learning are threefold, 1) the students get exposed to real-world challenges, 2) students get in-depth understanding of the presented material, and 3) hands-on exercises prepare students for careers in the industry. The labs uses open-source software to teach defensive security topics, how attackers attempt to bypass defences, and how to securely set up a couple of security features, such as Virtual Private Network (VPN) and Public Key Infrastructure with OpenSSL. Their research demonstrated that hands-on labs are highly efficient and effective in providing students with practical experience using the software tools covered in the labs, and gave the students a better understanding of the material presented. The majority of the students reported that the labs increased their interest in the subjects as well, thus increasing their self-interest of studying the topics on their own. [41] reported similar results, with students returning to the learning modules after completing them in class, often between the hours 18:00 and 21:00 on the same day or the following day. This indicates that students benefit from having the flexibility to access educational materials at their convenience, and from anywhere.

To address the limitations identified in existing local learning environments for cybersecurity, the thesis aims to introduce a robust and interactive learning environment designed to assist students in navigating the intricate landscape of template injection. Drawing inspiration from learning labs [10–13, 40], the project emphasises the need for a learning environment that not only provides challenges, but also guides learners through a progressive and comprehensive understanding of the vulnerabilities and their mitigation strategies. Striking a balance between XVWA's [11] brief explanation and WebGoats's [12] more detailed approach, the project will try to provide basic, but comprehensive learning material on the subject.

Taking cues from labs such as DVWA [10] that provide some forms of guidance, the learning environment will go a step further by actively offering student feedback through hints. This feature aims to enhance the learning experience by offering personalised guidance and valuable insights into the challenges, and facilitating a deeper understanding of the presented material.

In alignment with the findings of [41], where flexibility increased student engagement time, the challenges will be made available through the campus network for an extended period. This accessibility feature aims to encourage active student participation at their convenience, fostering a more dynamic learning experience.

### Online platforms

Another option for learning cybersecurity is through paid services that provide practical challenges, videos, and explanations on a wide array of security issues. Portswigger [42] offers all their labs and material for free, with lab challenges focusing on various web application vulnerabilities at different difficulty levels. These challenges are designed to be solved using their tool, Burp Suite, though some require the paid version of the tool.

Pentesterlab [7] and HackTheBox [43] offer a few free challenges, but reserve most behind a paywall. The extent of information provided about the security vulnerabilities varies, as it's the challenge authors' choice to determine what is necessary for the learner to know prior to it. Offensive Security [44] and Pentester Academy [8] provide structured courses with video tutorials and lab environments with objectives for participants to apply what they learn. Both offer this as a paid service as it is possible to get certified from their courses. Udemy [45] focuses on video education in a wide array of categories, everything from music to cybersecurity. While many courses are free, paid ones include certification, instructor messaging, and exercises to reinforce the material.

In contrast, OverTheWire [9], offers challenges where users must locate passwords for the next level within the current level. It provides links to helpful materials and a few hints on the challenge page, but OverTheWire is more of a practical playground than a traditional platform. Its challenges cover basic Unix commands, encoding, basic source code analysis for exploitation, common vulnerabilities and security misconfiguration, and reverse engineering.

Mitrovic et al. [46] extended SQL-Tutor [47] to assist university-level students in learning Structured Query Language (SQL) by providing feedback based on their errors. The authors investigated whether positive feedback would enhance learning more than only negative feedback would. Positive feedback is telling the participants which parts of their solution is correct, whilst negative feedback is telling when parts are wrong. Their results demonstrated that participants receiving both positive and negative feedback learned twice as fast as those receiving only negative feedback. The system offered hints divided into six levels, with level one indicating whether the submitted query is correct and level six providing the complete solution. Participants were given hints automatically on incorrect submissions until they reached level three, after which they had to actively request hints. The extended SQL-Tutor aimed to reduce uncertainty by highlighting the correct parts of the query. Mitrovic and the team's research highlighted that both negative and positive feedback contributed to faster learning and better comprehension. The combination of positive feedback along with negative feedback, further improved the learning process.

Research by Deng et al. [48] in dynamic cybersecurity learning environment has shown that tailoring the learning process for each individual student is beneficial for the learning outcome. The study analysed data from three local labs where students worked on their computers. The learning environment adjusted itself based on students' performance, skill levels, and predictions made by its prediction module. The implementation of a personalised learning environment resulted in improved learning performance, higher engagement, and increased student satisfaction. By utilising various data logs and monitoring students' time spent on different actions, the environment chooses an optimal learning style for each student. Additionally, instructors could identify at-risk students early, gain better insight into students' learning progress, and provide targeted assistance.

Notably, personalised labs had a more significant impact on high-performing students compared to low-performing ones.

Taking inspiration from the benefits of personalised learning highlighted by Deng et al. [48], the project will introduce adaptive generated hints. This tailored approach is anticipated to increase learning outcomes, engagement, and overall satisfaction. Building on the positive impact of gamification [35–38, 49], the learning environment will incorporate elements reminiscent of CTF challenges, transforming the educational experience into a 'treasure hunt' for the flag file.

Aligned with the findings of Mitrovic et al. [46], the system aims to provide participants with both positive and negative feedback with hints at varying levels, with focus on the negative feedback. This dual feedback approach has been shown to accelerate learning and improve comprehension. The system will strike a balance between guiding participants and allowing independent problem-solving.

Moreover, to accommodate diverse schedules, challenges will be accessible through the campus network over an extended period. This flexibility, aligned with the findings of [41] and [49], aims to increase participant engagement by allowing access at the students' convenience.

### 2.2.2 Feedback systems

As stated in [50], the purpose of feedback is to improve the students' self-efficacy, encourage perseverance and growth of the mindset, recognise accomplishments and foster active engagement. With this a change in behaviours, skills and knowledge of the student is possible by focusing on the experience and learning growth. A feedback system can then be defined as a system which aims to cultivate these aspects in a controlled environment through specific mechanisms.

Tiffany Barnes and John Stamper [51] conducted a study on automated feedback in a discrete mathematics course. The feedback was generated using Markov Decision Process (MDP) on historical student data. They observed that even for a new problem (a "cold start"), the system could still function as a problem-solving environment, and after one semester, it became capable of providing a significant number of hints. To optimise the system, the authors carefully balanced hint specificity and the amount of data used to create the MDP, maximising the value from the small dataset. This approach resulted in a system that could offer hints for over 80% of the cases. Additionally, they introduced multi-level hints, with the first-level hint providing information about sub-goals and the fourth-level hint offering specific details about how to achieve those sub-goals. Giving students hints for the next sub-goal and a correct solution has proven to enhance learning and skill transfer. Barnes and Stamper [51] suggested that grouping the students based on their skill level and providing different groups with tailored hints could be beneficial, advocating for further research in this area.

Lavbič et al. [17] developed and tested a SQL learning system that provided students with hints tailored to their individual problem-solving approaches, rather than guiding them toward a predefined ideal solution. This adaptive hint generation was facilitated by a prepopulated database containing solutions contributed by previous students. Over time, as more students use the system, the hint generator would become more versatile, supporting a wider range of solutions. This approach proved highly effective, particularly for students with limited prior knowledge of SQL. However, there are some drawbacks to such a system. It relies on a database of previous solutions, which can limit the ability for students to learn independently if the system isn't accessible remotely. Additionally, the students might find it challenging to deviate from the provided hints and enforce their own solutions. This study underscores the value of providing hints to students during the learning process, emphasising the importance of adapting hints to each student to maximise their learning support. Even with more basic hint-giving mechanisms, it's reasonable to believe that progressively offering hints throughout the learning process is beneficial.

Research done by Marwan et al. [18] in 2019 indicates that automated hints with textual explanations and self-explanations significantly improved performance by students in a programming course. The most significant improvements were observed when students tackled tasks similar to post-test objectives. Student feedback on the self-explanations of the hint was mixed. The purpose of the self-explanation prompts was to make the students reflect upon why the hints was given, to fortify the learning material. Their research suggests that students with low prior knowledge in the subject may have difficulty constructing meaningful explanations to why the hints were given. Meaning that low performing and new students in a topic would benefit more from not having self-explanation prompts on the hints. Even though these prompts could be confusing they do encourage students to take more time in reviewing the hints given, request fewer hints and follow more of the hints' content. Students given the prompts spent 64% more time viewing each hint than students without the prompts, and they asked for 67% as many hints. Marwan et al. [18] also recommended that students should be able to request hints at all times, as long as the hints were implemented in such a way that it could not be abused. One reason for their reasoning being that there did not seem to be an optimal time to provide hints as the timing varied from student to student.

Ruan et al. [21] found that offering hints which suggested specific code edits had no substantial negative impact on students' understanding compared to those who used test cases. In their research, they observed that students made an effort to comprehend why certain edits were recommended and how they contributed to solving the programming assignments, which led to a deeper understanding of the concepts. This suggests that providing hints with specific edits, while letting students figure out the reasoning behind these edits, may be more effective.

Hao et al. [52] also supported the idea of providing more detailed feedback. In their study, students could request unlimited amounts of hints, and interestingly, students who received less detailed hints tended to request more hints. This might be because the hints didn't help much, or students wanted confirmation of their code's correctness. However, they rarely seemed sure if they were on the right track. Furthermore, the research did not find any significant difference in academic performance between students who received hints and those who did not. This suggests that the value of feedback through auto-generated hints might be insignificant. The authors argued that this could be a result of only generating hints for the top five errors, potentially neglecting other common issues. Another reason could be that they did not provide multi-level hints due to concerns about potential exploitation by students. However, if hints were only provided after a specific duration of a certain number of attempts, multi-level hints might be a viable option. Nonetheless, Hao et al. [52] concluded that merely understanding the gap between the student's current code and the expected solution provided enough information for making progress and fixing errors. It's worth noting that this conclusion applies to contexts where students have multiple options of support alongside the system's feedback.

Malone et al. [49] employed gamification to increase student engagement and interest in a cybersecurity course using their system, Riposte. The system offers hints, feedback and personalised objectives based on students' performance and knowledge. By assessing their performance through pretests and monitoring progress through assignments, the system highlights their success, providing positive reinforcement feedback. To motivate students, the authors implemented a point system and leaderboard, awarding points upon reaching growth stretch goals. Riposte was initially used in a local on-campus environment, but adapted to an online format due to COVID-19. This transition provided students with greater flexibility, allowing them to work on the labs at their convenience for extended periods. The data collected by Riposte allowed instructors to track students' progress and intervene when needed. The integrated auto-grader in Riposte assigned a minimum expected grade based on submitted assignments. Combined with hints for improvements provided by the system, this approach encouraged students to continuously improve their assignments, boosting their engagement and motivation.

Research in a health informatics course by Alessandra Galassi and Pierpaolo Vittorini [15] examined whether allowing students to submit their assignments multiple times to receive feedback for iterative improvements had benefits. Most students in the study found the tool vital for exam preparation as it helped them understand their mistakes and the appropriate problem-solving methods. Indicating that automated and immediate feedback can enhance the learning process by consecutive highlighting misconceptions.

Price et al. [20] discovered that specifying which part of a student's submitted programming code the hints applied to was helpful. However, students often found these hints too vague because they lacked precise edits and explanations regarding why the edits were suggested. This was particularly problematic when the hints did not align with the student's solution.

The system developed by Švábenský was trained using a dataset of 13,446 shell commands, collected from various cybersecurity training sessions conducted by himself and other researchers [53]. In earlier work [54], Švábenský demonstrated that applying pattern mining to students' command-line history could unveil valuable insights into their problem-solving approaches, misconceptions, and areas of difficulty. This research also suggests that it's feasible to cluster students based on their behavioural patterns. The results indicate that clustering is well-suited for identifying learners' challenges, establishing correlations between learning behaviours and performance, and determining effective teaching strategies for each student. Additionally, the techniques utilised proved to be effective in identifying novel solutions. Clustering thus proved to be effective in identifying similarities and differences between approaches and behavioural patterns.

In a related study conducted by Švábenský [16], pattern matching was utilised to map students' progress onto two models: a reference graph and a milestone graph, which were used to provide post-training feedback to support students' reflection. These models were built based on a single correct solution provided by the instructor, without accounting for potential alternative solutions. The students were given access to a trainee graph, which attempted to align their commands with a reference graph, highlighting any incorrect commands and explaining the errors while also showing the correct path they eventually followed. The milestone graph tracks students' progress through a sequence of commands required to complete the exercises and records the number of attempts made by students at each step. Instructors utilise this tool to assess students' progress and understanding.

The milestone graph was inspired by work of Mirkovic et al. [55], who collected input and output data from students and mapped them to predefined milestones. Instructors utilised these milestones to monitor students' progress. The second model drew inspiration from the research of Andreolini et al. [56], who tracked students' progress in red team exercises via sub-goals and erroneous attempts, mapping the data to a reference graph and visualising the data in another graph. Švábenský [16] expanded on this mapping process to allow students progress in a non-linear fashion, using pattern matching to map students' commands to any state. This flexibility allowed students to achieve the end goal, even if they skipped certain steps. Švábenský [16] limited the work to only give visual feedback after each challenge, enabling students to see their own progression, identify areas where they made mistakes, and opening for post-training reflection. This also provided instructors with data to distinguish between low and high-performing students, enabling timely intervention and tutoring support. For remote students, this would require the instructor to contact each student to help them. Therefore the automated student feedback lessens the workload for the instructors as well as arriving in a timely manner. [23] discovered that regular and timely feedback improved the students' pass rates and average scores in a programming course. This highlights the positive impact of feedback, as it enhances performance, motivation, and comprehension of the subject matter. This is because the feedback allows the students to reflect on their understanding while the topic is fresh in mind. The authors also noted that allowing the students to retake tests throughout the course and providing feedback after each test made the students distribute their engagement more evenly throughout the course, reducing last-minute cramming near test deadlines.

Zheng et al. [57] explored the potential of enhancing hint generation of Large Language Model (LLM) models by iteratively refining and validating hints through a feedback mechanism. This approach employs a chain-of-thought methodology to enhance the models' awareness of the context in which they generate hints. Their results indicate that this technique significantly improves the accuracy of hints and performance across various benchmarks, leading OpenAI's Generative Pre-trained Transformer (GPT) to outperform several other larger models.

Maciej Pankiewicz and Ryan S. Baker [58] researched whether OpenAI's GPT could be used for automatically generating personalised hints for students' programming assignments, thereby alleviating the demanding task of providing tailored feedback for instructors. The assignments covered basic object-oriented programming concepts in C#. Their research indicates that the generated hints were beneficial; the students who received them did use significantly less time to solve assignments, even when the hints were disabled. There were indications that students might become overly dependent on the GPT-generated hints, but the issue seemed to correct itself rapidly as the students kept working without the GPT-generated hints. Despite the hints being generated in Polish by GPT, which could have had an effect on the quality and effect of the hints, the results were positive and promising.

In Švábenský's latest work [14], the system underwent further testing of the automated formative feedback post-training and its impact on the learning. The researchers proposed that utilising data generated by learners could alleviate the burden on instructors to give individual feedback to each student by automating feedback delivery. They also sought answers to the questions of how and when it is best to provide students with feedback to enhance learning, with a specific emphasis on guiding learners on their next steps. An identified challenge is the difficulty in automatically addressing all relevant factors for each challenge, potentially resulting in generated feedback that may not be equally relevant for every student. The authors argue that timely and personalised feedback is crucial for learning, particularly in remote education where instructors may have limited awareness of students' activity. The authors plan to conduct future studies involving two groups—one receiving this automated feedback and the other not—to assess its effectiveness.

Inspired by the milestone approach of Mirkovic et al. [55], as expanded upon and visualised in Švábenský's work [16], and [51], the learning material will adopt the same approximation in the process of discovering and exploiting injection vulnerabilities. The hint generation and the detail level will also be influenced by this approach. Drawing inspiration from the utilisation of command-line history [53, 54], the hint algorithm will consider the injected payloads of the participants.

Emphasising the benefits highlighted Švábenský [16] and Lavbič et al. [17], tailored hints can enhance learning outcomes, increase engagement, and overall satisfaction. The thesis aims to focus on the assistance these hints can provide for students and instructors. In addition to being immediate, this feedback can significantly enhance the learning process, as indicated by Marwan et al. [18]. Such feedback allows students to understand their mistakes promptly, promoting iterative improvements. While detailed edits in feedback have been seen to be beneficial and lead to deeper understanding [21], the planned learning environment is intended to gradually increase the detail level of hints. This approach aims to encourage students to research the challenge topic themselves before receiving more detailed hints, avoiding the provision of full-fledged solutions to the challenges. Providing students with a solution might lead them to copy-paste the solution, obtain the flag, and proceed to the next challenge without understanding why the solution worked, especially in the context of a CTF competition where time is limited. Additionally, as the learning environment is designed for individuals with limited prior knowledge of the subject, providing tailored hints at a level that does not give away a complete solution is deemed more effective, aligning with the approach stated in [17]. This approach supports individual problem-solving.

While this thesis will not utilise clustering for behavioural pattern analysis as in [54], instructors can identify areas of misconception and difficulty by analysing user log data for repetitive injections and interactions. This data will also pave the way for future project expansion to incorporate clustering and behavioural pattern analysis. This expansion will enable grouping students based on their skill level, allowing for customised hints on a group basis, as suggested by Barnes and

Stamper [51]. This grouping could facilitate better priming and hint restrictions between each group, enabling more effective fine-tuning of hints.

This thesis will neither utilise the progressive hint generation technique developed by Zheng et al. [57]. The reason being that this would significantly increase hint generation time and cost. Since timely delivery of hints have been proven important [14, 23], this technique will not be used until the speed of such technique is improved. However, the thesis aims to leverage a LLM model as GPT to generate personalised instructor-like hints through data-driven prompt engineering, as demonstrated in [58].

However, unlike much research this thesis aims to utilise LLM to automatically generate instructor-like hints. Taking notes from [52], certain criteria will assist the provision of hints to the learner's situation.

Table 2.1 below shows an overview of the most important related work mentioned in this chapter, which field of research and their capabilities and features with regards to this thesis.

**Table 2.1:** An overview of the main related works from the litterateur study, and their capabilities

| Related work | Year | Field | Capabilities and Features |
|---|---|---|---|
| [48] | 2018 | Computer Science | Personalised labs, monitor student progression |
| [23] | 2018 | Programming | Automated feedback and marking, feedback pre-defined, identification of student misconceptions |
| [18] | 2019 | Programming | Automated data-driven next-step hints with textual explanation |
| [21] | 2019 | Programming | Personalised hint generation, visualising programs |
| [52] | 2019 | Programming | Automated formative feedback, hints for top 5 issues |
| [28] | 2020 | Cybersecurity | Monitor user progression, automated and on-going assessment of student's work |
| [34] | 2020 | Cybersecurity | Educational goal displayed per challenge, monitor user progression, goals and sub-goals as game scenarios, storytelling elements |
| [50] | 2021 | Programming | Automated feedback on assignments |
| [15] | 2021 | Programming | Automated feedback, pre-defined feedback |
| [20] | 2021 | Programming | Data-driven hint generation on programming assignments |
| [16] | 2022 | Offensive cybersecurity | Automated post-training feedback |
| [58] | 2023 | Programming | Data-driven personalised hint generation |
| [14] | 2023 | Offensive cybersecurity | Added visualised command-line history and error analysis post-training to [16] |

## 2.3 Technical Background

In this section, we will provide the necessary technical background to facilitate understanding of the thesis and to introduce the algorithms used in the research.

### 2.3.1 Generative Artificial Intelligence

AI refers to systems which mimic human intelligence and possess the capability to predict and generate content [59, 60]. In the field of Natural Language Processing (NLP), the transformer architecture has emerged as a pivotal innovation, revolutionising language understanding and setting new benchmarks in natural language processing tasks [61–64].

Transformers belong to a class of models that leverage self-attention mechanisms to assess the significance of individual components within an input and capture the relationships between them. They have found extensive application in NLP and computer vision tasks [65]. Typically, transformers adopt an encoder-decoder architecture. The encoder processes input data and produces a meaningful representation, while the decoder employs this representation to generate an output [61]. Transformer models typically undergo pre-training on massive datasets, followed by fine-tuning for specific tasks. Pre-traning is typically unsupervised learning, where the model learns patterns and language structures from the input data. By making pre-trained models available, researchers can reduce their computational requirements, contributing to a more environmentally friendly research by enabling more researchers to utilise these models in their own work. Fine-tuning, is then performed to adapt the model for a specific task, such as translating English into French.

Generative AI, built on neural network techniques like transformers, generative adversarial networks and variational autoencoders, is designed to create new content by recognising patterns and structures in existing data [63]. It aims to generate novel outputs based on the knowledge acquired during training. LLM, a subtype of generative AI, excels in producing natural-sounding text [62]. Generative AI models can be trained in various ways, including unsupervised and semi-supervised methods, enabling them to learn from vast amounts of unlabeled data during pre-training, which, in turn, supports further research. Notable examples of such models include OpenAI's GPT-4 and DALL-E 2, and Google's BERT.

LLMs, a type of subtype of transformers, excel in understanding language and generating text. They are specifically designed to process large volumes of text, comprehend semantic meanings, grammar, and contextual nuances within the analysed text. These models can have a vast number of parameters, with some reaching into the billions, excel at capturing intricate language patterns. Parameters being characteristics of the input text. LLMs are highly proficient in transfer learning, meaning that pre-training them on diverse datasets will subsequently enable them to be fine-tuned for various language tasks, such as translation, question answering, and text classification [66]. Part of this learning process involves tuning the reward model, which assesses the quality of the model's outputs, allowing for adjustments to yield improved results in subsequent iterations [67].

### GPT 3.5

GPT-3, a member of the generative transformer-based LLM family developed by OpenAI, is a significant milestone in the progression of language processing models [68]. It is widely recognised for its application in Chat-GPT. With 175 billion parameters, GPT-3 can handle texts of up to 2048 tokens, making it an ideal choice for advanced natural language processing tasks. However, it's important to note that it does require substantial computational resources for optimal performance. GPT-3.5 employs a layered attention mechanism, akin to the sparse transformer model, restricting token attention to on their locally nearest token-neighbours. This approach enhances computational efficiency by reducing complexity [69, 70].

The model's training process involved pre-training on a vast dataset of 570GB of text data from the Internet, with a knowledge cut-off point in September 2021 [69, 71]. Following this, GPT-3.5 was fine-tuned to perform language-related tasks, encompassing translation, text summarisation, and question answering. Due to this extensive training, the model performs well in zero-shot and few-shot learning scenarios. Zero-shot learning capabilities enables the model to perform tasks it has never encountered before, relying solely on textual task description. Few-shot learning capabilities, enables the model to perform new tasks with just a few examples or instructions [69].

GPT-3.5 incorporated Reinforcement Learning from Human Preferences (RLHF) into its algorithm. RLHF leverages human feedback on the model's output to further improve its performance. Compared to its predecessor, GPT-3.5 boasts fewer parameters, reducing its resource-intensive nature and enhancing practicality [72]. OpenAI has also published an optimised variant of GPT-3.5 for, tailored for chat functionality, GPT-3.5-turbo [73]. This version further refines the model's capabilities in conversational contexts and text generation.

## 2.3.2 Classifiers

### Cosine Similarity

The cosine similarity algorithm calculates the similarity between two vectors by measuring the cosine of the angle between them. The lower the cosine between them, the more similar the vectors are. This is often used to measure similarity of text. An advantage of this algorithm is that two vectors can be quite similar even though their size differs, which is not the case for other comparison algorithms such as Levensthein distance and Euclidean distance. The algorithm projects the objects to be compared into a multi-dimensional space and compares the cosine of the angle between them, where each word is one dimension. The occurrence of a word determines its magnitude, but it does not change the angle between vectors. The calculation considers only non-zero coordinates, which helps reduce the algorithm's complexity [74].

### Levensthein Distance

Levensthein distance calculates how different two strings are; the higher the number, the more different they are. The algorithm measures the difference by calculating how many changes are needed to change one string into the other. A change can be changing a character into another, such as "a" into "k", deletion of a character or insertion of a new character [75].

# Chapter 3

# Methodology

This chapter provides an overview of the thesis methodology, including the selection of literature, development cycle of the project, and the process of gathering and processing data. Insights from structured interviews with cybersecurity experts and the literature review played a pivotal role in shaping the design of the learning environment.

The thesis adapted the design science research method, extensively used for applied research in educational research [16]. The design science research method splits the research process into five steps [76], as illustrated in Figure 3.1. The method can be categorised into two phases: the development of the artefact (the system or software) and its evaluation. Evaluation leads to refinement of the artefact and subsequent round of evaluation. This cyclic process continues until the allocated research time ends or until the artefact demonstrates sufficient utility and evidence that it addresses the targeted problem.

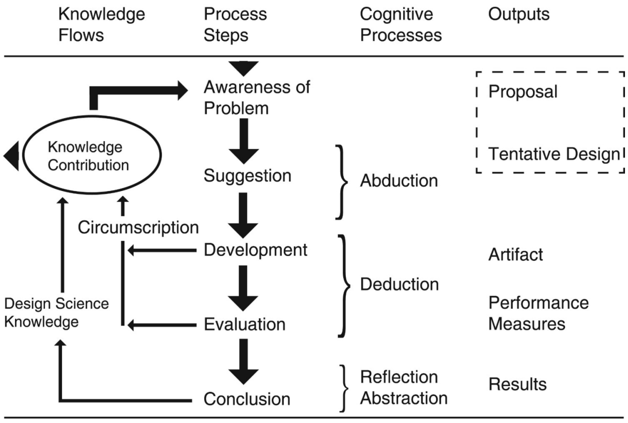


**Figure 3.1:** The processes in the design cycle [76]

## 3.1 Awareness of Problem

Awareness of the problem was explored through a literature study and structured interviews of cybersecurity experts. Both contributed to the suggestion of the artefact design. The literature study was conducted to acquire a comprehensive understanding of the current state of the art within cybersecurity education. Studies were selected for their valuable insights applicable to cybersecurity education. Some were chosen due to their relevance to specific aspects that align with this thesis, while others provided a broader view of the field. Their ideas and results inspired and formed this thesis. The literature was sourced from theses, articles, and scientific databases.

A structured interview was conducted with experts gathered at the cyber exercise Locked Shields 2023 in April. Five individuals were interviewed, including various team leads and members of different red teams. These experts possess several years of experience in finding and developing exploits, as well as implementing mitigating measures for said exploits. The purpose of these interviews was to inspire and guide the direction of the project. The seven questions from the interview can be found in Appendix A.

## 3.2 Suggestion

The development of the artefact was undertaken using a combination of Python Flask, a web framework, and HTML, which formed the core of the artefact. Flask was responsible for rendering the web-based application, while SQLAlchemy facilitated the storage and management of database tasks. These technologies were selected for their efficiency, flexibility, and suitability for the artefact's requirements. The focus on template injection in this thesis arises from the recognition that using template frameworks to create dynamic web pages is often the most efficient and beginner-friendly approach. Consequently, it is widely adopted and needs to be secured. Moreover, since the artefact's core utilises Flask, implementing this vulnerability was straightforward, providing a means to test the overall structure of the learning environment and its effectiveness. Additionally, HTML templates were employed for the user interface. To ensure a cohesive and functional design, HTML templates were sourced from [77] and further adapted to align seamlessly with artefact's unique functionality. Allowing for visually appealing and user-friendly user experience.

The artefact comprises of several components (see Figure 3.2):

- Front-end web application: This component serves as the user interface where the learners interact with the system. It is implemented in HTML, CSS, and JavaScript.
- Hint provider: Responsible for generating and delivering hints to the users.

- Back-end database: Stores data related to hints, solutions, and users.
- Logger: Records user interactions within the challenges.
- Questionnaire: Collects users' perceived experience.

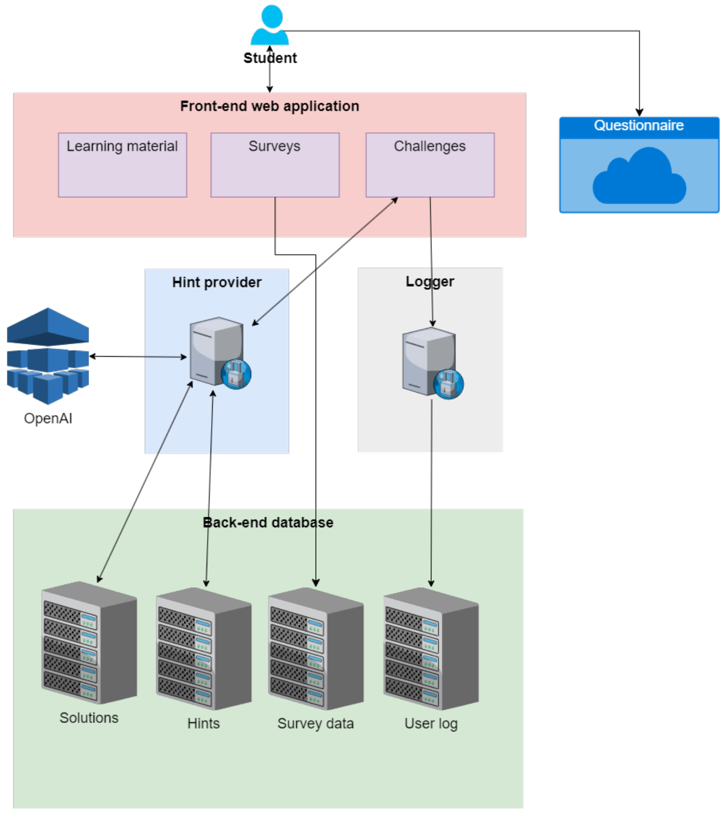


**Figure 3.2:** System architecture

### Front-end web application

The front-end web application component will serve the users with surveys, learning material and the challenges, including the display of hints. Surveys to map knowledge before and after completing the lab challenges. Learning material covering the process of identifying and exploiting injection vulnerabilities. Testing the acquired knowledge from the learning material through the hands-on lab challenges. Within the challenges is where the assisting hints will be displayed.

### Hint provider

The hint provider component is designed to generate hints and deliver hints to users, track hints given to each user, and record new solutions. It generates hints based on pre-made prompts and on-demand during the challenges based on the users' inject and the closest solution. The hints provided will be tailored to the user's progress in the exploit process and progress since last hint. Additionally, when the flag string is found in the rendered webpage, the hint provider checks if the last inject exists in the database as a solution. If not, the inject is recorded for future use in generating hints for other users.

### Back-end database

The back-end database will store all known possible solutions to the challenges, generated hints, survey answers, and the users' activity inside the challenges. Recorded user activity include: page browsing, injected payloads, time, and hints given.

### Logger

The logger will record the users' interaction within the challenges, which page they browse, what they try to inject, and when they perform these actions.

### Questionnaire

After the lab exercises the user will be asked to complete a questionnaire about their user experience. The questionnaire will be adjusted based on the test group and whether the user received hints or not. The questionnaires themselves will be hosted on an external platform, *nettskjema.no*.

## 3.3 Development

An initial alpha build of the artefact (excluding the hint provider) was developed using Python Flask. Observations and feedback from the pilot test were instrumental in refining the artefact and laying the groundwork for the hint provider.

An exploratory focus group, comprising of volunteer students from the third-year at the Norwegian Defence Cyber Academy, was used to test the artefact. During this phase, observations of the artefact's flow, user interactions and the students' feedback were recorded to inform further refinements.

Based on the results from the exploratory test and feedback from instructors, several enhancements were made. These include the implementation of pre-survey and post-survey functionalities within the artefact. To facilitate online student participation and random allocation of test groups, automatic distribution of credentials replaced handwritten notes. Furthermore, the artefact was designed to link pre-survey and post-survey responses to the same username. To streamline the user experience, a redirection to the correct questionnaire after the post-survey was implemented, reducing the likelihood of students accessing the wrong version.

## 3.4 Evaluation

Assessing the effectiveness of generated hints were accomplished by comparing data gathered from the focus groups, which were randomly split into two groups, one receiving hints and one control group without. Utilising benchmarks used in research by Marwan et al. [18], differences between the groups were measured using the Kruskal-Wallis test, with number of injects replacing the number of hints, as hints are directly related to injects and represent a variable in both groups. The Kruskal-Wallis test is preferred as it does not require the normality of data distribution Additionally, Spearman's correlation was employed to investigate any significant correlation between completed objectives, number of injects sent and time used. Spearman's correlation evaluates the strength and direction between two variables, determining if one variable increases or decreases with the other, and it does not require linear data [78]. The same variables were analysed using ANOVA tests, following Rivers [22], to investigate whether hints had any significant impact on some of these. Additionally, surveys were employed to gauge students' knowledge before and after interacting with the artefact (see Appendix C).

Understanding students' perceptions of the learning process is crucial for assessing the effectiveness of the feedback. Therefore, Welch's one-way ANOVA analysis of responses from structured questionnaires (see Appendix B), designed to measure experiences and perspectives — a method inspired by Beckman et al. [79], was used. Welch's one-way ANOVA was utilised due to potential unequal variance among the groups. The significance level was set to 0.05 for all tests, following common practice [80].

Data preparation, including parsing and preprocessing, was conducted using custom-written Python scripts. Subsequently, statistical analysis was carried out using JASP [81] and Jamovi [82], versatile statistical software programs offering a range of analytical tools.

# Chapter 4

# Design and Implementation

This chapter will explain the design, flow, and implementation of the artefact, as well as explaining why some technology was chosen. An overview of the artefact's workflow can be seen in Figure 4.1.

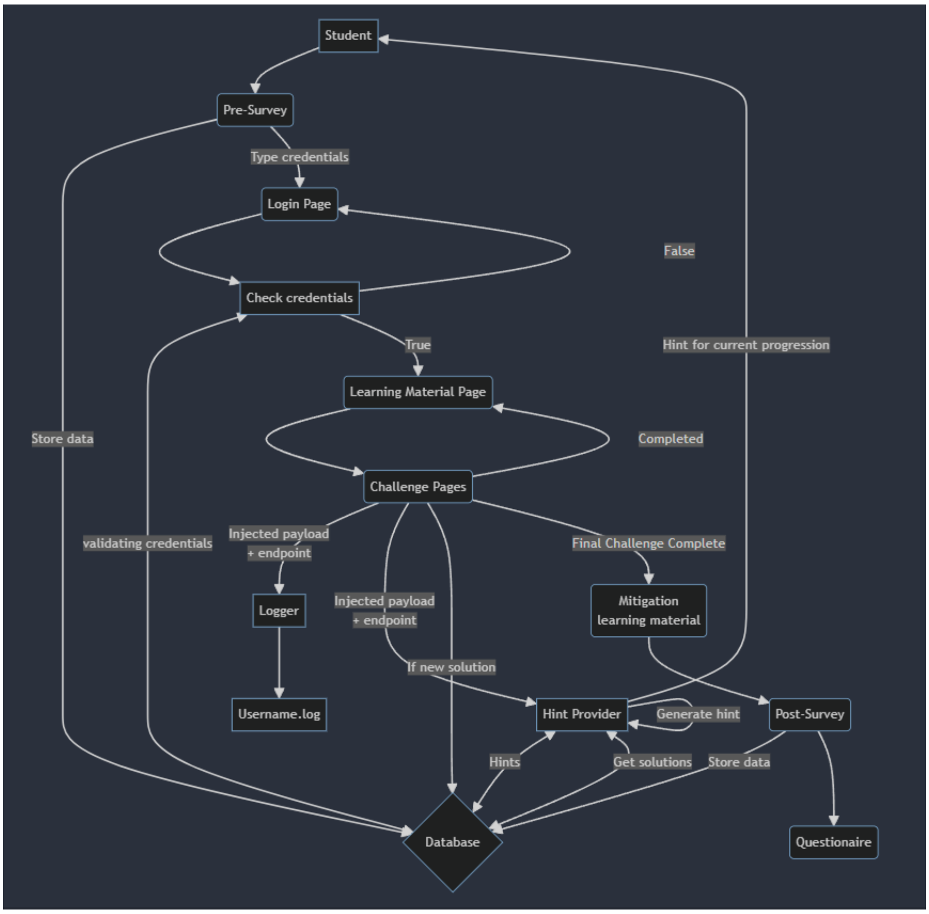
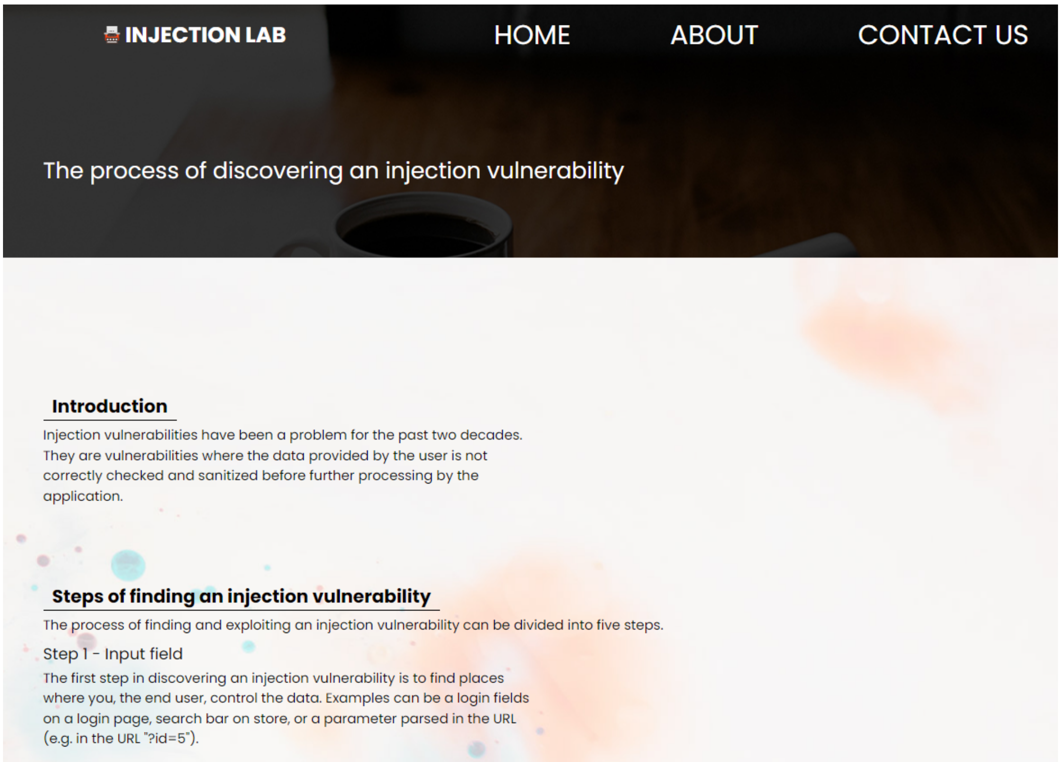


**Figure 4.1:** Overview of the artefact's workflow

Upon accessing the artefact, participants were required to fill out a short pre-survey gauging their prior knowledge of certain web security concepts. Subsequently, they received their credentials and were redirected to the login page. Following this, they were guided to learning material that covered the process of identifying and exploiting injection vulnerabilities (Figure 4.2).



The process of discovering an injection vulnerability

Introduction

Injection vulnerabilities have been a problem for the past two decades. They are vulnerabilities where the data provided by the user is not correctly checked and sanitized before further processing by the application.

Steps of finding an injection vulnerability

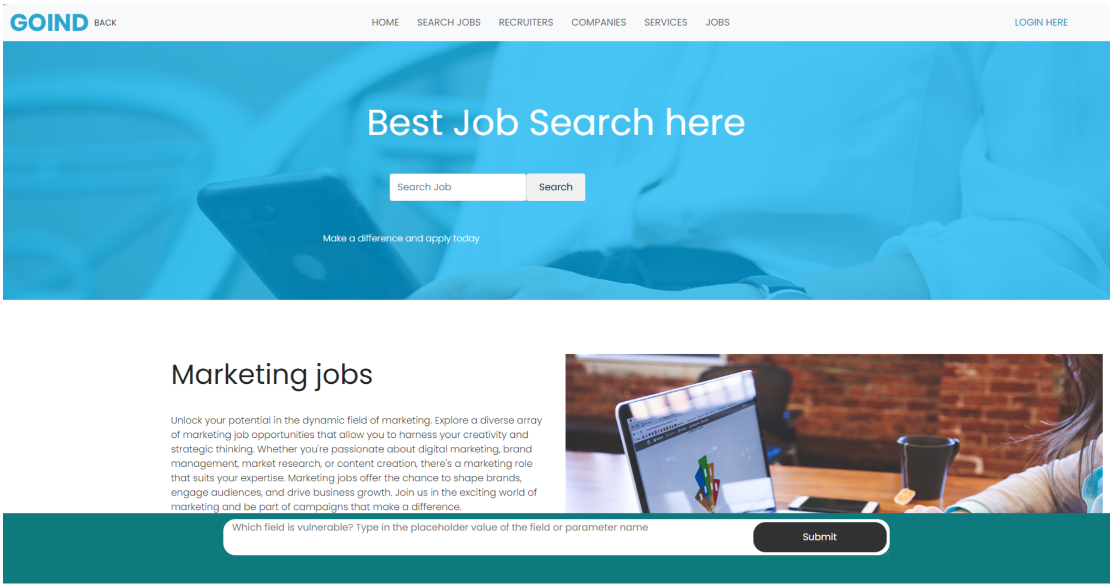
The process of finding and exploiting an injection vulnerability can be divided into five steps.

Step 1 - Input field

The first step in discovering an injection vulnerability is to find places where you, the end user, control the data. Examples can be a login fields on a login page, search bar on store, or a parameter parsed in the URL (e.g. in the URL "?id=5").

**Figure 4.2:** Initial learning material

After reviewing the learning material, users could access the challenges (Figure 4.3). During the challenges, the artefact logged the users' attempted injects and where. Based on the users' injection attempts, the hint provider sent hints to assist them in completing the challenge (appearance of a hint can be seen in Figure 4.4).



Best Job Search here

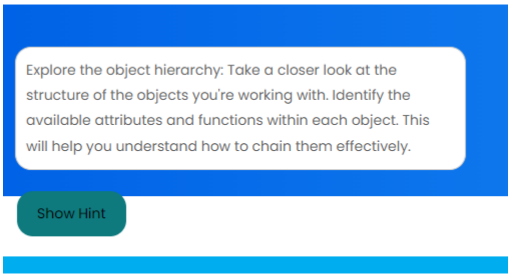
Make a difference and apply today

Marketing jobs

Unlock your potential in the dynamic field of marketing. Explore a diverse array of marketing job opportunities that allow you to harness your creativity and strategic thinking. Whether you're passionate about digital marketing, brand management, market research, or content creation, there's a marketing role that suits your expertise. Marketing jobs offer the chance to shape brands, engage audiences, and drive business growth. Join us in the exciting world of marketing and be part of campaigns that make a difference.

Which field is vulnerable? Type in the placeholder value of the field or parameter name

**Figure 4.3:** User interface inside a challenge



Explore the object hierarchy: Take a closer look at the structure of the objects you're working with. Identify the available attributes and functions within each object. This will help you understand how to chain them effectively.

**Figure 4.4:** Displaying a hint

After engaging with the artefact, users received learning material about mitigating measures against injection vulnerabilities. Subsequently, they were prompted to complete a post-survey to assess their learning outcomes. Upon survey submission, users were directed to a questionnaire (see Appendix B), where they were asked to rate their experience with the artefact in regards to various aspects using a 5-point Likert scale. Additionally, users had the option to provide free-text feedback.

### Hint provider

When initialising the artefact for the first time, it creates a pool of hints that include both automatically generated hints from OpenAI's GPT-3.5-turbo and human-pre-written hints. The selection of GPT-3.5-turbo for automated hint generation was grounded in its exceptional capabilities. Given its advanced natural language processing proficiency in formulating guiding hints, it is well-suited for generating hints that align with the requirements of CTF-like exercises. Specifically, its ability to understand and contextualise user queries, coupled with its expansive knowledge base, positions it as a robust tool for crafting hints tailored to the challenges presented in CTF-like exercises. The timely delivery of hints is crucial for student guidance [23], and GPT-3.5-turbo's responsiveness matches the dynamic nature of the learning process. Its user-friendly Application Programming Interface (API) simplifies integration into the artefact. Additionally, GPT-3.5-turbo offers a cost-effective solution for hint generation. The decision not to choose the next version, GPT-4, was driven by considerations of increased cost and its limit of 50 messages every 3 hours [83].

The hints are generated based on prompts which are designed for each step of the exploitation process, specifying the hint detail level they will provide. Meanwhile, the pre-written hints were crafted using data from the pilot group, where the hint provider was not included. These hints consider the user's progress in the exploitation process, their performance level, and factors these into setting step and hint detail level the injected payload must meet for the hint to be applicable. In cases where no hints could be found for the current inject or all stored hints were used, a new hint would be generated and given to the student (Listing 4.1).

**Code listing 4.1:** Example of a prompt

```
fr"""In an ethical lab environment I developed for cyber security
     education I need to provide the students with some hints to
    help them. A student have given this as an inject '{inject}'
    and the closest possible solution is '{closest}'. Could you
    provide a detailed next-step hint to help the student get a
    step closer to the provided possible solution in this
    template injection challenge? Do not mention the closest
    possible solution and give the hint as Hint: """
```

**Code listing 4.2:** Example of a premade prompt

```
In an ethical lab environment I developed for cyber security
    education I need to provide the students with some hints to
    help them understand the chain of objects can be called.
    Could you provide some detailed hints for template injection
    to help them with this in the context of a Jinja2 without
    access to the files? Give the hints in the format as I would
    give directly to the student.
```

The detail level of the hints are categorised into three levels. The first level is relatively generic, providing guidance on the current step of the exploitation process. At detail level two, the hints suggest methods to help the students start a step-wise approach toward the solution. Detail level three instructs students on specific actions required to make the inject work, such as using encoding, bypassing filtering, or employing other means.

At the highest hint detail level, the algorithm randomly selects an applicable, unused hint from the database or generates a new hint based on the latest injected payload and its most similar valid solution. The similarity check uses a combination of *cosine similarity* and *Levenshtein distance*. Cosine similarity was chosen for it ability to compare text without regards to the frequency of occurrence, which is not a factor for injection payloads. By decomposing injections and solutions into single character the algorithm is able to effectively compare the injects to all solutions. And its low complexity makes it efficient and easy to implement. Levenshtein distance was chosen for it simplicity and how it calculates similarity differently from the cosine similarity. The Levenshtein distance is normalised to preventing it from heavily outweighing the cosine similarity, which produces values between 0 and 1. This normalised Levenshtein distance is then added to the Cosine similarity value. This evaluation is performed for all stored solutions, and the solution yielding the highest value is chosen as the basis for the hint. This customisation ensures that each student receives a hint tailored for their inject.

The hints given each to user are stored in the database to prevent hint repetition. Hints are provided after every fourth inject, with increasing detail if the user is on the same step of the exploitation process as when the last hint was provided. When the completion criteria is met, the artefact checks if the student's answer is one of the stored solutions. If not, it is added to the database of possible solutions, contributing to generating hints for other students later on.

### Deployment

Two versions of the artefact were deployed: one providing hints, and the other without hints. All components, except the questionnaire, were housed within a Docker container, one container for each version. To accommodate for automatic distribution of credentials a third Docker container was utilised, also handling the surveys and redirected students to the correct questionnaire after the post-survey. The deployment was carried out using Docker Compose, exclusively within NTNU's cyber range. This restricted access to students and teachers attending the school, aiming to safeguard the artefact and its knowledge from unauthorised use for unethical purposes.

# Chapter 5

# Evaluation

This chapter presents and discusses the results obtained through the methods outlined in Chapter 3, covering an analysis of collected log data, questionnaire responses, and surveys. The log data represents the performance of students during their interaction with the artefact. The questionnaire responses offer valuable insights into the students' learning experiences, while the surveys track students' progression from before to after, allowing us to gauge their learning outcomes.

The results are analysed separately per test group, followed by a discussion of how these findings contribute to answering each research question. The synthesis of these results addresses the main research question. Towards the end of the chapter, ethical considerations and limitations of the research are also presented.

## 5.1 Data Analysis

An overview of the number of participants and questionnaire responses per test group can be seen in Table 5.1.

**Table 5.1:** Data overview

| Test group | Questionnaire responses/Number of participants |
|---|---|
| The Norwegian Defence Cyber Academy students | 15/17 |
| NTNU students | 13/32 |
| CTF participants | 1/50 |

### 5.1.1 The Norwegian Defence Cyber Academy results

The exploratory focus group consisted of 17 third-year students from the Norwegian Defence Cyber Academy, with eight randomly assigned to the hint group and nine to the control group. Notably, only one student, from the control group, completed all challenges. In compliance with the Armed Forces requirements for research, all data had to be deleted after the project's conclusion, and students had to provide their consent by filling out a consent form (see Appendix D) before participating. During testing, a technical error in challenge 3 was discovered for the control group. Consequently, data related to challenge 3 (time spent, number of inject, and completed objectives) was excluded from the analysis for both groups to ensure a valid data basis.

### User logs

Analysing the data with the Kruskal-Wallis test revealed no significant difference between the same variable in the groups, as seen in Table 5.2.

**Table 5.2:** Kruskal-Wallis test comparison between the Norwegian Defence Cyber Academy groups

| Measured variable | p |
|---|---|
| Total minutes | 0,386 |
| Objectives completed | 0,346 |
| Total injects | 0,564 |

Comparing the variables between the groups through the ANOVA test, no significant difference were found, as shown in Table 5.3. The results of these two tests suggest that the provision of hints did not significantly affect any of the recorded data.

**Table 5.3:** ANOVA comparison of the Norwegian Defence Cyber Academy groups

| Measured variable | p |
|---|---|
| Total minutes | 0,342 |
| Objectives completed | 0,362 |
| Total injects | 0,530 |

Table 5.4 displays the results of Spearman's correlation of the variables within each group, as well as for all students. Unfortunately, analysis of *objectives completed* was not possible for the hint group due to absence of variance in the variable, since all students completed the same number of objectives. This was a consequence of having a small number objectives and having to delete data related to challenge 3, which further limited the data. However, the comparison found a significant correlation between *total minutes* and *total injects* within the control group and across all the data. This observation aligns with the intuitive expectation that more time allows for more attempted injects. Additionally, the correlation between *objectives completed* and *total minutes* were almost significant, which aligns with the expected result that the longer students try, the more they achieve.

**Table 5.4:** Spearman's correlation of variables the Norwegian Defence Cyber Academy groups

| **Compared variables** | $p_{all}$ | $p_{hint}$ | $p_{control}$ |
|---|---|---|---|
| Total minutes - Total injects | 0,002 | 0,078 | 0,021 |
| Objectives completed - Total injects | 0,621 | - | 0,702 |
| Objectives completed - Total time | 0,055 | - | 0,114 |

### Questionnaire

Out of the 17 participants, 15 answered the post-test questionnaire. A Welch's one-way ANOVA analysis of the overlapping questions between the groups revealed no significant difference, as seen in Table 5.5. Indicating that the experience with the artefact was fairly similar, and the hint did not significantly impact other aspects of the artefact.

**Table 5.5:** Welch's one-way ANOVA comparison of Norwegian Defence Cyber Academy questionnaire data

| **Question** | **p** |
|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | 0,772 |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | 0,834 |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 0,525 |
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) | 0,522 |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) | 0,927 |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) | 0,722 |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) | 0,406 |

In their written feedback, some students reported that they felt they learned a lot about injections. Overall, the feedback was neutral, with a slightly positive perception of the learning environment and a slightly negative perception of the generated hints. While the students generally regarded the artefact positively, there was a consensus that the last challenge posed a too considerable challenge.

Most participants found the initial two challenges manageable, but they encountered a significant difficulty spike with the last challenge, partly due to having to skip the third challenge. Some expressed a wish for more intermediate challenges between challenge 2 and 4, due to this spike, particularly from the control group. Consequently, students felt a lack of prerequisites for handling the final challenge, which is also reflected in the number who successfully completed it (only one). Despite issues with the third challenge, participants acknowledged that it was a reasonable foundation for the lead up to the final challenge. However, there should have been more emphasis on the fact that the challenges required following the entire step-by-step approach, starting with rediscovering the vulnerable field for each challenge. Some students struggled to initiate inject-building for the last challenge after identifying the vulnerable field. Despite the difficulty spike in the last challenge, some students found it exciting to tackle.

It was challenging for students to ascertain if they were on the right path, indicating that the learning material may not be sufficient to enable them to gauge their progress accurately.

In general, the hints were reported as subpar; they need more optimisation and fine-tuning to be more applicable for each task the students are working on to be truly helpful and credible, especially for those with little prior knowledge of SSTI. As hints were provided after a number of injects, some students found it challenging to progress without them. This suggests that students made progress because of the hints, even though a lack of prior knowledge rendered many hints confusing and unhelpful. Additionally, all students in the control group strongly agreed that hints would have been helpful in tackling the challenges.

Students found the step-wise approach to injection vulnerabilities helpful. The examples of different template engine syntax were also reported as beneficial, but they expressed a desire for examples of SSTI code execution. Overall, the students found the learning material to be well-written and helpful, but an even more thorough introduction could have been beneficial.

Since the overall response to the last challenge was that it was too difficult, one character ("_") was removed from the blacklist filter before the next test at NTNU. This adjustment aimed to simplify the process of bypassing the filter, eliminating the need for the students to encode and encapsulated their injection payloads for a successful injection. Additionally, a pre-survey and post-survey, as well as the automatic distribution of credentials were implemented before the next test.

**Table 5.6:** Average ratings of the Norwegian Defence Cyber Academy questionnaire data

| **Question** | **Avg**$_{Hint}$ | **Avg**$_{control}$ |
|---|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | 4,143 | 4,25 |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | 3,857 | 3,75 |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 3,714 | 3,375 |
| Were the frequency and quantity of hints sufficient to assist in your progress with the challenges? (1: Strongly disagree, 5: Strongly agree) | 2,857 | NA |
| Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree) | 2,714 | NA |
| Did the content of the hints facilitate steady progress with the challenges without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree) | 3 | NA |
| Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | 1,857 | NA |
| Was the exercises challenging? (1: Not at all, 5: Extremely challenging) | NA | 4,142 |
| Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree) | NA | 3,429 |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | NA | 4,143 |
| Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much) | NA | 3,286 |
| Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful) | NA | 5 |
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) | 3,714 | 3,5 |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) | 3,286 | 3,25 |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) | 3,857 | 3,25 |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) | 3,571 | 4 |

### 5.1.2 NTNU results

The confirmatory focus group consisted of 32 participants, composed of undergraduate and graduate students majoring in computer science at NTNU. 15 were randomly assigned to the hint group, while the remaining 17 formed the control group. Participants hailed from both second and third years, predominantly from the programming course, with a few from the data course. Although programming and security knowledge were generally similar between the two courses, there is a knowledge gap corresponding to the academic year. The participants were familiar with C or C++, with minimal knowledge of Java and JavaScript. Some had Python knowledge, and a few had experience with GO. All students attended the training session voluntary, and their decision to participate did not impact their course grades. The test took place during a regular school day, allowing both online and on-campus students at NTNU Gjøvik to participate. The artefact remained available for an extended period after the allocated lab time to observe whether students would return; however, no participants revisited it after the first day.

#### User logs

Analysing the data with the Kruskal-Wallis test to identify any significant difference between the groups and their variables did not reveal any, as shown in Table 5.7. Similarly, the ANOVA test did not indicate any significant difference either, as shown in Table 5.8. This suggests that providing hints did not have a significant impact on the students.

**Table 5.7:** Kruskal-Wallis test comparison between NTNU groups

| Measured variable | p |
|---|---|
| Total minutes | 0,427 |
| Objectives completed | 0,272 |
| Total injects | 0,088 |

**Table 5.8:** ANOVA comparison between NTNU groups

| Measured variable | p |
|---|---|
| Total minutes | 0,313 |
| Objectives completed | 0,230 |
| Total injects | 0,086 |

The results of Spearman's correlation are presented in Table 5.9. These results revealed a significant correlation between all variables analysed. Strong correlations were observed between all variables when analysing the entire dataset, as well as a strong correlation between *objectives completed* and *total injects* within each group. These results are unsurprisingly, as more time spent will allow for more injects, and more injects could lead to building inject payload which works, resulting in completing more objectives. The findings would have been significant if not all correlations were significant.

**Table 5.9:** Spearman's correlation of variables in NTNU test groups

| **Compared variables** | $\mathrm{p}_{all}$ | $\mathrm{p}_{hint}$ | $\mathrm{p}_{control}$ |
|---|---|---|---|
| Total minutes - Total injects | <,001 | 0,010 | 0,023 |
| Objectives completed - Total injects | <,001 | <,001 | <,001 |
| Objectives completed - Total time | <,001 | 0,017 | 0,041 |

### Questionnaire

For the questionnaires, only four participants from the hint group and nine from the control group provided answers. Running their responses through Welch's one-way ANOVA did not reveal any significant difference in the user experience between the groups, as shown in Table 5.10. Indicating that on this limited data the hint did not have a significant impact.

**Table 5.10:** Welch's one-way ANOVA comparison of NTNU questionnaire data

| Question | p |
|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | 0,381 |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | 0,329 |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 0,788 |
| Was the exercises challenging? (1: Not at all, 5: Extremely challenging) | 0,873 |
| Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree) | 0,676 |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | 0,135 |
| Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much) | 0,609 |
| Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful) | 0,646 |
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) | 0,290 |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) | 0,249 |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) | 0,673 |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) | 0,123 |

The learning environment received mixed feedback, with some positive aspects noted. However, areas of improvement were identified as well, such as making the placeholder text in the submission field clearer. Furthermore, there is still potential for improvements in structuring the learning material clearly into the different steps and establish a stronger foundation for beginners, as it was perceived as too general.

The overall response to the hints was slightly negative. Students who received hints didn't always find them applicable to the specific aspect of the exploit that they were trying to manipulate. The hints were perceived as not very helpful in aiding the students' steady progress to complete challenges, and, consequently, they didn't significantly impact their sense of accomplishment. Notably, the hint group reported a greater sense of satisfaction and accomplishment than the control group, with average scores of 4 and 3.125, respectively.

The quality and frequency of the hints were reported to be neither particularly good nor bad. Furthermore, the hint group found that the exercises reflected the learning material better than the control group (3.8 and 3.5), suggesting that the hints contributed to understanding the learning material. However, the control group overall rated the learning material more positively, as seen in Table 5.11.

The feedback suggests that the hints need further refinement and control mechanisms to ensure that information they provide is accurate and appropriately tailored. In one instance, a solution to the final challenges was inadvertently given in a hint for the first challenge. Students from the control group found the last challenge quite challenging, feeling uncertain about how to start or progress; they felt there were too many alternatives to try. Conversely, one student in the hint group initially faced challenges starting the first challenge, but found the subsequent challenges manageable. The hint group rated the difficulty of the lab slightly higher than the control group (average score of 3,8 and 3,375 respectively), indicating that the hints could have posed an additional challenge in understanding injection vulnerabilities. However, when asked, "*Was the exercises challenging?*" the hint group's ratings were slightly lower than the control group (3.4 and 3.5 respectively). This aspect should be investigated further with more challenges and participants.

Removing certain characters from the input with the blacklist filter unintentionally led to the last two challenges seeming to use a different template engine, both Express Language and Jinja2 became possible alternatives. This unintended outcome made the challenges harder than intended.

**Table 5.11:** Average ratings of the NTNU questionnaire data

| **Question** | **Avg**$_{Hint}$ | **Avg**$_{control}$ |
|---|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | 3,8 | 3,375 |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | 2,5 | 3,375 |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 2,6 | 2,75 |
| Were the frequency and quantity of hints sufficient to assist in your progress with the challenges? (1: Strongly disagree, 5: Strongly agree) | 2,25 | NA |
| Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree) | 2,25 | NA |
| Did the content of the hints facilitate steady progress with the challenges without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree) | 2,25 | NA |
| Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | 2,25 | NA |
| Was the exercises challenging? (1: Not at all, 5: Extremely challenging) | 3,4 | 3,5 |
| Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree) | 3 | 2,75 |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | 4 | 3,125 |
| Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much) | 3,8 | 3,5 |
| Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful) | 4,2 | 3,875 |
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) | 3 | 3,625 |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) | 3 | 3,625 |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) | 3,4 | 3,625 |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) | 2,6 | 3,625 |

### Survey

Due to technical issues with the artefact, the post-survey had to be completed using an external form rather than within the system. This circumstance might have resulted in fewer responses and introduced uncertainties regarding which username and group corresponds to each answer. In the post-survey, three responses were received from the hint group and five from the control group. Interestingly, one user managed to submit the post-survey through the learning environment, while the rest did not. Due to this, the analysis will focus only on the data from the eight participants who answered both the pre-survey and the post-survey. The ANOVA test revealed no significant difference in the amount of correct answers from before to after going through the lab, as shown in Table 5.12. However, it's noteworthy that all students improved their scores from the pre-survey to the post-survey.

**Table 5.12:** ANOVA analysis of NTNU students' survey data

| **Correct answers** | **p** | **$\text{Avg}_{Hint}$** | **$\text{Avg}_{Control}$** |
|---|---|---|---|
| Pre-survey | 0,818 | 2,333 | 2,2 |
| Post-survey | 0,957 | 5,667 | 5,6 |

Based on the responses in the survey, it seems that students' confidence regarding SSTI has slightly increased as a result of the exposure to the vulnerability type through the artefact, as shown in Table 5.13. But due to the limited data and uncertainties about whether the user responses are related to the correct group, the findings are regarded as inconclusive and require further research.

**Table 5.13:** NTNU students' confidence in handling template injection

| **Question** | **Option 1** | **Option 2** | **Option 3** | **Option 4** |
|---|---|---|---|---|
| Pre: What is your experience with offensive security and template injection vulnerabilities? | None<br>2<br>2 | Basic<br>3<br>1 | Intermediate | Advanced |
| Pre: How confident are you in your ability to detect injection vulnerabilities? | Not confident<br>1<br>2 | Slightly<br>4<br>1 | Moderately | Very |
| Post: How would you rate your understanding of template injection vulnerabilities? | Poor | Fair<br>3<br>2 | Good<br>1<br>1 | Excellent |

Black = control group, Blue = hint group

### 5.1.3 CTF competition results

The last challenge was featured in a CTF competition at NTNU, where hints were provided to all participants. Initially, 50 participants accessed the challenge; however, after excluding those who spent less than five minutes, only 16 remained. It's worth noting that the actual number might be even lower, as suggested by the user logs that some users injected directly into the vulnerable field from the start. This could be a result of participants deleting their cookies, utilising automated tools or scripts, or receiving guidance from other participants on how to proceed. Following the competition, participants were asked to fill out a questionnaire regarding their experience with the template injection challenge and hints provided.

#### User logs

Since the challenge was deployed with hints for all users, there will not be any groupings, making it impossible to run ANOVA analysis on the user logs. However, Spearman's correlation of the measured variables is still possible. Since some user logs completed the challenge in under five minutes, the Spearman's correlation was run on all users that spent more than one minute and on those who spent more than five minutes. The results of these comparisons are shown in Table 5.14. In both cases there was observed a strong correlation between *total minutes* and *total injects*, not surprising, as more time allows for more injects.

**Table 5.14:** Spearman's correlation between variables for CTF users

| **Compared variables** | $\mathbf{p}_{1min}$ | $\mathbf{p}_{5min}$ |
|---|---|---|
| Total minutes - Total injects | 0,096 | 0,196 |
| Objectives completed - Total injects | 0,004 | 0,034 |
| Objectives completed - Total time | 0,615 | 0,729 |

**Questionnaire**

There was only one participant who answered the questionnaire after the CTF competition. The feedback was overall negative regarding the challenge and hints; this version of the last challenge did not fit the context of a CTF competition. For a CTF competition, it might have been better to run the first version of challenge 4 instead of the one which was tested at NTNU. Including "_" in the blacklist would have pushed participants to figure out a way to bypass this restriction. As participants in CTF competitions often are cybersecurity enthusiasts, the challenge, as it stood, appeared too easy. Additionally, having multiple pages and fields, which made it harder to locate the vulnerable field, was not received positively either because it required more exploration of possible vulnerable input fields.

## 5.2 Discussion

> *What benefits does providing feedback in the form of generated hints during the injection exploitation process, in the context of learning aid beginners in learning its concepts?*

The goal of this research question was to examine the benefits that generated hints could offer novice individuals in their learning journey, specifically in understanding injection flaws. The overarching goal is to empower cybersecurity professionals with the skills to recognise, protect against, and mitigate these vulnerabilities. Subsequent exploration of sub-questions will evaluate the efficiency, perceived usefulness, and potential improvements of the generated hints, contributing to a comprehensive understanding of their impact on the learning experience.

### 5.2.1 Research Question 1

> *Did the hints contribute to less time and attempts spent on the challenges?*

Average time difference between the Norwegian Defence Cyber Academy and NTNU is likely influenced by the fact that the Norwegian Defence Cyber Academy test was conducted after school hours, allowing for more time, while the NTNU test was conducted in between courses on a school day, as well as the removal of "_" from the blacklist filter in challenge 3 and 4 which made the challenges easier for the students at NTNU. Because of this the comparison will just consider within focus groups data, and not cross test group comparison.

Analysing the data from the Norwegian Defence Cyber Academy, no significant difference was found between the two groups. Upon examination of the average time spent and average number of inject, it appears that hints may have had a slight negative impact, as both the time spent and number of inject for the hint group are slightly higher. However, this difference is not statistically significant. These results could have been affected by the network issues the test initially had, where some students could access the lab while others couldn't. The issue was resolved after approximately 30 minutes. Additionally, there was a technical issue with challenge 3 which made the challenge impossible to solve for the control group, thus invalidating the related data. After removing data related to challenge 3, no significant results were found on the ANOVA test (Table 5.3) or difference in significant correlations on the Spearman's correlation (Table 5.4).

In the NTNU data, with the adjustment of the last two challenges and a working challenge 3 for both groups, the differences have changed. In this case, the hint group had lower averages, sending 35.3% less injects compared to the control group. On average, they also spent about 12 minutes (16.95%) less time on the challenges, suggesting that they might have devoted more time to consider their injects. Despite these observations, no significant difference was found between the groups with a ANOVA test (Table 5.8), and the groups exhibited the same significant correlations in the Spearman's correlation (Table 5.9). It's worth noting that some students may have quit after the technical issues with the survey at the start of the lab, possibly getting deterred that the rest of the artefact was of the same quality, rendering some of the data non-representative.

**Table 5.15:** Average time spent and average number injects

| **Test group** | **Time**$_{avg}$ | **Injects**$_{avg}$ |
|---|---|---|
| $TheNorwegianDefenceCyberAcademy_{hint}$ | 125,35min | 171,75 |
| $TheNorwegianDefenceCyberAcademy_{control}$ | 111,31min | 151,78 |
| $NTNU_{hint}$ | 61,40min | 103,21 |
| $NTNU_{control}$ | 73,93min | 159,53 |

In conclusion to research question 1: hints seemed to affect the number of injects, but only slightly affect the time spent. Due to technical issues, no definitive conclusions can be drawn, but the results seem to suggest that the hints may contribute to less injects and time spent.

### 5.2.2 Research Question 2

*How useful is the feedback in improving students' knowledge?*

For the students at the Norwegian Defence Cyber Academy, the hints were received slightly negatively. The frequency, quality and applicability were perceived to be moderately adequate (average ratings between 2,71 and 2,86). The hints were reported to not significantly negatively affect the students' sense of satisfaction and accomplishment. This could be a consequence of the hints not being especially helpful, as they in some cases addressed aspects which some students already had handled or not tackling the specific problem the students were facing. The lack of prior knowledge made some hints confusing as they used terms related to template injection which the students were not familiar with. Despite the hints not being very helpful, one student reported that it was challenging to progress without hints. This might indicate that the frequency of the hints was not sufficient for novices. In the control group, everyone strongly agreed that hints would have been helpful during the challenges.

The results from the test at NTNU indicated that the students found the hints to be less helpful (with average ratings between 2,25 and 2,75). The students at NTNU also wished to receive hints on a higher frequency, as well as feeling that the hints were not applicable to the aspect of the inject they were trying to manipulate. The hints did not significantly assist in their progress on the challenges. However, the students reported a lesser sense of satisfaction and accomplishment because of the hints compared to the students at the Norwegian Defence Cyber Academy. This is evident when one student reported that a hint received on challenge 1 contained a solution for challenge 4, thereby disrupting the learning process of building the inject himself. Due to the technical issues with the test, there were not many answers on the questionnaire, making the data not representative for the group, as only 33% of this focus group answered.

In conclusion to research question 2: the hints were not peculiarly helpful. The survey results from the NTNU students showed improvement from before to after the interaction, but this improvement was observed in both groups, suggesting that the improved scores were not correlated to the hints. The hints were more confusing and distracting. Indicating that for the hints to be helpful the hint generator needs more fine-tuning and the hint provider needs better provision criteria controls.

### 5.2.3 Research Question 3

*How can the feedback be improved?*

Considering the feedback from the students, there is a suggestion to provide hints at higher frequency. As mentioned in [14, 16, 23, 28, 48], timely feedback is important for learning, but the optimal timing may vary among students [18], therefore hints could be provided upon request from the students themselves with a restriction to avoid abuse of hints, a restriction such as in the work of Marwan et al. [18]. Using simpler language in hints, avoiding complex technical terms related to template injection, could enhance understanding, making the hints more useful for students. Comprehensibility and the applicability of the hints could be improved by designing better prompts that consider these parameters, among others. The generation and provision of hints could be based on all the injects of the student since the last hint, so that the hint comments on the student's approach rather than only their latest inject. To ensure that the feedback does not contain a full-fledged solution, which takes away the sense of accomplishment and satisfaction of completing a challenge, restrictions on the hints should be implemented. To further improve the hints, a wider range of positive feedback can also be implemented, as positive feedback has been shown to be beneficial for learning [46].

In summary of research question 3, enhancing the feedback system could involve introducing hints on-demand with restrictions to hinder abuse, conducting a more comprehensive analysis of students' injection timeline, simpler language, and incorporating more positive feedback. Additionally, the learning material could be improved to increase the comprehension of template injection concepts and terms, thereby making it easier to understand the feedback.

To answer the main research question, based on the conducted research covered in this thesis, it appears that providing generated hints during the exploitation process did not significantly affect the learning outcomes. The survey data indicates that both groups improved to a similar extent. Notably, the hint group at NTNU utilised substantially fewer injects during their interaction with the artefact. This could be a result of their thoughtful evaluation of what and how to design their injects, but it could also be because the hints introduced another challenge by confusing and distracting them. Additionally, there is a possibility that the students in the hint group exited the lab earlier than some of the students in the control group. This seems most likely since, on average, the control group completed 2,65 challenges, while the hint group completed 2 challenges. The results suggest that there were no discernible benefits of having hints during the exploitation process. However, it cannot be definitively concluded from this study due to the limited data, along with uncertainties regarding its reliability and validity.

## 5.3 Ethical Considerations and Limitations

Considering whether it is ethical to educate students about concepts which could be exploited for nefarious means, the artefact uses information and technology which is publicly available. As mentioned earlier, *knowledge itself is not inherently good or bad; it is the intention of the person utilising the knowledge which is*. And as Sun Tzu emphasised [27], understanding the tactics and techniques employed by the enemy is crucial to effectively react and defend oneself. Considering these perspectives, there should be no concerns with educating individuals about these concepts.

Concerns have been raised regarding the power and water consumption of large generative AI models and their environmental impacts [84]. Water is used for cooling the equipment that runs these models, with models like GPT consuming about 500 milliliters when asked five to 50 questions [85]. In order to minimise the environmental impact, the artefact primarily requests hints at initialisation and stores them for later use. This approach enables reuse of the same hints for multiple students without the need to repeatedly ask the model to generate hints.

The study has several limitations. Firstly, as mentioned earlier, the tests have all been voluntary. For students at Norwegian Defence Cyber Academy, it was held after school hours and limited to third year students, potentially reducing the number of potential participants as the students did have other activities and homework to attend to. For the students at NTNU, it was held in the middle of the day as part of a voluntary lab in a course. These students had other courses both before and after the lab, and since the lab did not impact their grades, some might have chosen alternative assignments or activities. At NTNU, it should have been emphasised and stressed repeatedly that the artefact would be left available for several days to possibly increase student engagement.

Secondly, the survey and questionnaire data is limited due to technical issues encountered during testing. The total number of participants across all tests is also limited, but the participants represent the target demographic of the artefact well. These issues impacted the data's validity and reliability as students had to manually navigate to the post-survey and correct questionnaire. In the post-survey, the students had to recall their own usernames, introducing an element of uncertainty regarding correct recollection and whether the artefact recorded the pre-survey data under the same username. Regarding the questionnaire, two different versions were made for each group, and students were required to fill out the one corresponding to their allocated group, hint, or control group. Based on the written feedback, it's evident that some students accessed the wrong version. While it's possible that more students did the same, it was not apparent from the written feedback.

The hint provider does not take into account the student's injection history or identify specific areas within the student's inject that require attention to facilitate progress. Instead, it considers which step in the exploitation process the student is at, the hint detail level, the latest inject, and its most similar solution.

During the implementation, the number challenges were limited and restricted to SSTI in favour of the hint provider algorithm. The small number of challenges, combined with a small sample size, has made it impossible to assess whether hints had a significant impact on the students' learning.

An unforeseen consequence of the blacklist filter removing certain characters from the input made it appear as if the last two challenges could be using another template engine. This resulted in confusion among some students and made the challenges more difficult than intended, potentially causing frustration when the syntax for the believed engine did not work.

# Chapter 6

# Conclusion and Future Work

## 6.1 Conclusion

Firstly, an artefact was developed, encompassing learning material, hands-on exercises, and automated feedback in the form of hints. To evaluate the hints effect, comparison of user log data of students who received hints and a control group without hints were analysed. The gathered data, encompassing number of injects, time spent, and objectives completed, was compared using Kruskal-Wallis test(refer to Table 5.2 and Table 5.7) and ANOVA test (refer to Table 5.3 and Table 5.8). Additionally, Spearman's correlation was employed to investigate any significant correlations between variables within each group. The results from these algorithms found no significance in the data.

Secondly, students at NTNU filled out surveys mapping their knowledge prior to and after interacting with the artefact. The results show that the automated hints did not significantly affect the results. On the other hand, just interacting with the artefact did increase the students' scores.

Thirdly, after each test, participants were asked to complete a questionnaire containing questions about their experience regarding different aspects of the artefact. The obtained results underwent analysis using Welch's one-way ANOVA tests for the overlapping questions (refer to Table 5.5 and Table 5.10), which revealed no significant differences. Overall, the challenges and hints were perceived slightly negatively. However, students in the control group reported that they thought hints would have been helpful, indicating that the suggested artefact had a desirable feature, just not optimally implemented.

This study was conducted to find out whether automated hint generation during hands-on offensive security exercises would provide any benefits to the learning process. The experiment was organised so that the effect of these hints could be observed between two groups, one with and one without. Due to technical issues that resulted in a limited dataset with uncertain validity and reliability, the results of the thesis are inconclusive. Further testing on a larger test group is needed to properly evaluate the artefact.

## 6.2 Future Work

Regarding future work there are several areas that could be explored. The experiments should be performed again to gather valid and reliable data, and assess the artefact based on these. Furthermore, fine-tuning and optimising of the hint provider could increase the effectiveness of the hints. The hint generation could benefit from having a greater adaptability to the students' current inject, as well as having more restrictions on the content of the hints. The provision of hints could benefit from considering additional criteria for selecting hints, such as identifying the part of the inject the student is currently addressing based on the injects since the last hint. Involving instructors for hint quality assurance could further enhance the hints, ensuring a more instructor-like quality.

Although this thesis did not include an array of feedback techniques, future studies should research the prospect of implementing more, such as more elaborate positive feedback, which has been shown to have a positive impact on learning [46]. Tell the students when they are building the injection correctly and when they have fixed an issue they seem to have been struggling with. Furthermore, inclusion of the post-training visualisation graphs introduced in [16] would enhance students' post-training reflection. This would also enable instructors to easily identify common misconceptions among the students. Expanding on the visualisation, it's worth exploring the possibility of incorporating a graphical view during exercises. This view could display all the injects the student have attempted, breaking down injects into smaller parts and utilising coloured graph nodes to indicate whether an inject is a dead-end or can be further explored.

Future research could explore alternative models for generating hints that might outperform the current one and, ideally, be freely accessible. This investigation could further lower the entry barrier for utilising the artefact. An ideal model would excel in formulating technical hints based on prompts and possess the ability to generate a diverse array of hint types.

The last suggestion for future work in this study is to implement more challenges and other sub-categories of injection vulnerabilities, stretching the difficulty curve and widening the scope of the artefact. With this it will become easier to assess the effect of the generated hints since there will be a greater possibility of difference between the test groups.

# Bibliography

[1] hahwul. (2021), [Online]. Available: https://www.hahwul.com/cullinan/history-of-owasp-top-10/ (visited on 02/04/2023).

[2] S. Christey, B. Martin, M. Brown and A. Paller. '2009 cwe/sans top 25 most dangerous programming errors.' (2009), [Online]. Available: https://cwe.mitre.org/top25/archive/2009/2009_cwe_sans_top25.html (visited on 19/10/2023).

[3] M. Corporation. '2022 cwe top 25 most dangerous software weaknesses.' (2023), [Online]. Available: https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html (visited on 19/10/2023).

[4] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt and R. Karri, *Asleep at the keyboard? assessing the security of github copilot's code contributions*, 2021. arXiv: 2108.09293 [cs.CR].

[5] J. Umawing. 'Chatgpt writes insecure code.' (2023), [Online]. Available: https://www.malwarebytes.com/blog/news/2023/04/chatgpt-creates-not-so-secure-code-study-finds (visited on 20/05/2023).

[6] M. Liu, K. Li and T. Chen, 'Deepsqli: Deep semantic learning for testing sql injection,' in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2020, Virtual Event, USA: Association for Computing Machinery, 2020, pp. 286–297, ISBN: 9781450380089. DOI: 10.1145/3395363.3397375. [Online]. Available: https://doi.org/10.1145/3395363.3397375.

[7] PentesterLab. (2022), [Online]. Available: https://pentesterlab.com (visited on 10/05/2022).

[8] PentesterAcademy. (2022), [Online]. Available: https://www.pentesteracademy.com (visited on 10/05/2022).

[9] *Natas*, 2022. [Online]. Available: https://overthewire.org/wargames/natas/.

[10] digininja, *Damn vulnerable web application (dvwa)*, version 2.0.1, 2022. [Online]. Available: https://dvwa.co.uk.

[11] S. Thomas and SaMaN, *Xtreme vulnerable web application (xvwa*, 2020. [Online]. Available: https://github.com/s4n7h0/xvwa.

[12] O. Foundation, *Owasp webgoat*, version 8.2.2, 2022. [Online]. Available: https://owasp.org/www-project-webgoat/.

[13] J. Druin, *Owasp mutillidae ii*, version 2.8.78, 2022. [Online]. Available: https://github.com/webpwnized/mutillidae.

[14] V. Švábenský, J. Vykopal, P. Čeleda and J. Dovjak, 'Automated feedback for participants of hands-on cybersecurity training,' *Education and Information Technologies*, 2023. DOI: https://doi.org/10.1007/s10639-023-12265-8.

[15] A. Galassi and P. Vittorini, 'Automated feedback to students in data science assignments: Improved implementation and results,' in *CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter*, ser. CHItaly '21, Bolzano, Italy: Association for Computing Machinery, 2021, ISBN: 9781450389778. DOI: 10.1145/3464385.3464387. [Online]. Available: https://doi.org/10.1145/3464385.3464387.

[16] V. Švábenský, 'Automated feedback for cybersecurity training,' Ph.D. dissertation, Masaryk University Faculty of Informatics, Feb. 2022.

[17] D. Lavbič, T. Matek and A. Zrnec, 'Recommender system for learning sql using hints,' *Interactive Learning Environments*, vol. 25, no. 8, pp. 1048–1064, 2017. DOI: 10.1080/10494820.2016.1244084. [Online]. Available: https://doi.org/10.1080/10494820.2016.1244084.

[18] S. Marwan, J. Jay Williams and T. Price, 'An evaluation of the impact of automated programming hints on performance and learning,' in *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ser. ICER '19, Toronto ON, Canada: Association for Computing Machinery, 2019, pp. 61–70, ISBN: 9781450361859. DOI: 10.1145/3291279.3339420. [Online]. Available: https://doi.org/10.1145/3291279.3339420.

[19] J. P. Bernius, S. Krusche and B. Bruegge, 'Machine learning based feedback on textual student answers in large courses,' *Computers and Education: Artificial Intelligence*, vol. 3, p. 100 081, 2022, ISSN: 2666-920X. DOI: https://doi.org/10.1016/j.caeai.2022.100081. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X22000364.

[20] T. W. Price, S. Marwan and J. J. Williams, 'Exploring design choices in data-driven hints for python programming homework,' in *Proceedings of the Eighth ACM Conference on Learning @ Scale*, ser. L@S '21, Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 283–286, ISBN: 9781450382151. DOI: 10.1145/3430895.3460159. [Online]. Available: https://doi.org/10.1145/3430895.3460159.

[21] R. Reis, G. Soares, M. Mongiovi and W. L. Andrade, 'Evaluating feedback tools in introductory programming classes,' in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–7. DOI: 10.1109/FIE43999.2019.9028418.

[22] K. Rivers, 'Automated Data-Driven Hint Generation for Learning Programming,' Jul. 2017. DOI: 10.1184/R1/6714911.v1. [Online]. Available: https://kilthub.cmu.edu/articles/thesis/Automated_Data-Driven_Hint_Generation_for_Learning_Programming/6714911.

[23] P. Rattadilok and C. Roadknight, 'Improving student's engagement through the use of learning modules, instantaneous feedback and automated marking,' in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 802–806. DOI: 10.1109/TALE.2018.8615306.

[24] R. Shen. 'Injection to magecart.' (2020), [Online]. Available: https://www.akamai.com/blog/security/web-application-and-api-protection-from-sql-injection-to-magecart (visited on 26/03/2022).

[25] O. Foundation. 'Owasp top ten.' (2021), [Online]. Available: https://owasp.org/www-project-top-ten/ (visited on 26/03/2022).

[26] Tines. 'Voice of the soc.' (2023), [Online]. Available: https://www.tines.com/reports/voice-of-the-soc-2023 (visited on 27/10/2023).

[27] S. Tzu and S. B. Griffith, *The art of war*. Oxford, England: Clarendon Press, 1964.

[28] J. Mirkovic, A. Aggarwal, D. Weinman, P. Lepe, J. Mache and R. Weiss, 'Using terminal histories to monitor student progress on hands-on exercises,' in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20, Portland, OR, USA: Association for Computing Machinery, 2020, pp. 866–872, ISBN: 9781450367936. DOI: 10.1145/3328778.3366935. [Online]. Available: https://doi.org/10.1145/3328778.3366935.

[29] S. Fulton, D. Schweitzer and J. C. Dressler, 'What are we teaching in cyber competitions?' *2012 Frontiers in Education Conference Proceedings*, pp. 1–5, 2012.

[30] L. McDaniel, E. Talvi and B. Hay, 'Capture the flag as cyber security introduction,' in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 5479–5486. DOI: 10.1109/HICSS.2016.677.

[31] M. hendrix, A. Al-Sherbaz and V. Bloom, 'Game based cyber security training: Are serious games suitable for cyber security training?' *International Journal of Serious Games*, vol. 3, Mar. 2016. DOI: 10.17083/ijsg.v3i1.107.

[32] M. Beltrán, M. Calvo and S. González, 'Experiences using capture the flag competitions to introduce gamification in undergraduate computer security labs,' in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 574–579. DOI: 10.1109/CSCI46756.2018.00116.

[33] D. Gaurav, Y. Kaushik, S. Supraja, A. Khandelwal, K. Negi, M. Prasad Gupta and M. Chaturvedi, 'Cybersecurity training for web applications through serious games,' in *2021 IEEE International Conference on Engineering, Technology Education (TALE)*, 2021, pp. 390–398. DOI: 10.1109/TALE52509.2021.9678531.

[34] S. Karagiannis and E. Magkos, 'Adapting ctf challenges into virtual cybersecurity learning environments,' *Information and Computer Security*, Nov. 2020. DOI: 10.1108/ICS-04-2019-0050.

[35] J. Hamari, J. Koivisto and H. Sarsa, 'Does gamification work? — a literature review of empirical studies on gamification,' Jan. 2014. DOI: 10.1109/HICSS.2014.377.

[36] C. Dichev and D. Dicheva, 'Gamifying education: What is known, what is believed and what remains uncertain: A critical review,' *International Journal of Educational Technology in Higher Education*, vol. 14, Dec. 2017. DOI: 10.1186/s41239-017-0042-5.

[37] M. Sailer and L. Homner, 'The gamification of learning: A meta-analysis,' *Educational Psychology Review*, vol. 32, pp. 77–112, 2019.

[38] M. E. Morales-Trujillo and G. A. García-Mireles, 'Gamification and sql: An empirical study on student performance in a database course,' *ACM Trans. Comput. Educ.*, vol. 21, no. 1, Dec. 2021. DOI: 10.1145/3427597. [Online]. Available: https://doi.org/10.1145/3427597.

[39] A.-A. O. Affia, A. Nolte and R. Matulevičius, 'Integrating hackathons into an online cybersecurity course,' in *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2022, pp. 134–145.

[40] X. Wang, Y. Bai and G. C. Hembroff, 'Hands-on exercises for it security education,' in *Proceedings of the 16th Annual Conference on Information Technology Education*, ser. SIGITE '15, Chicago, Illinois, USA: Association for Computing Machinery, 2015, pp. 161–166, ISBN: 9781450338356. DOI: 10.1145/2808006.2808023. [Online]. Available: https://doi.org/10.1145/2808006.2808023.

[41] W. Halimi, C. Salzmann and D. Gillet, 'Access to massive open online labs through a mooc,' in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, ser. L@S '17, Cambridge, Massachusetts, USA: Association for Computing Machinery, 2017, pp. 331–334, ISBN: 9781450344500. DOI: 10.1145/3051457.3054017. [Online]. Available: https://doi.org/10.1145/3051457.3054017.

[42] P. Ltd. (2022), [Online]. Available: http://portswigger.net (visited on 10/05/2022).

[43] HackTheBox. (2022), [Online]. Available: https://www.hackthebox.com (visited on 10/05/2022).

[44] O. Security. (2022), [Online]. Available: https://www.offensive-security.com (visited on 12/05/2022).

[45] I. Udemy. (2022), [Online]. Available: https://www.udemy.com (visited on 12/05/2022).

[46] A. Mitrovic, S. Ohlsson and D. Barrow, 'The effect of positive feedback in a constraint-based intelligent tutoring system,' *Computers & Education*, vol. 60, pp. 264–272, Jan. 2013. DOI: 10.1016/j.compedu.2012.07.002.

[47] A. Mitrovic, 'Learning sql with a computerized tutor,' *SIGCSE Bull.*, vol. 30, no. 1, pp. 307–311, Mar. 1998, ISSN: 0097-8418. DOI: 10.1145/274790.274318. [Online]. Available: https://doi.org/10.1145/274790.274318.

[48] Y. Deng, D. Lu, C.-J. Chung, D. Huang and Z. Zeng, 'Personalized learning in a virtual hands-on lab platform for computer science education,' in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–8. DOI: 10.1109/FIE.2018.8659291.

[49] M. Malone, Y. Wang and F. Monrose, 'An online gamified learning platform for teaching cybersecurity and more,' in *Proceedings of the 22nd Annual Conference on Information Technology Education*, ser. SIGITE '21, SnowBird, UT, USA: Association for Computing Machinery, 2021, pp. 29–34, ISBN: 9781450383554. DOI: 10.1145/3450329.3476859. [Online]. Available: https://doi.org/10.1145/3450329.3476859.

[50] S. H. Edwards, 'Automated feedback, the next generation: Designing learning experiences,' in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21, Virtual Event, USA: Association for Computing Machinery, 2021, pp. 610–611, ISBN: 9781450380621. DOI: 10.1145/3408877.3437225. [Online]. Available: https://doi.org/10.1145/3408877.3437225.

[51] T. Barnes and J. Stamper, 'Toward automatic hint generation for logic proof tutoring using historical student data,' in *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, ser. ITS '08, Montreal, Canada: Springer-Verlag, 2008, pp. 373–382, ISBN: 9783540691303. DOI: 10.1007/978-3-540-69132-7_41. [Online]. Available: https://doi.org/10.1007/978-3-540-69132-7_41.

[52] Q. Hao, J. P. Wilson, C. Ottaway, N. Iriumi, K. Arakawa and D. H. Smith, 'Investigating the essential of meaningful automated formative feedback for programming assignments,' in *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2019, pp. 151–155. DOI: 10.1109/VLHCC.2019.8818922.

[53] V. Švábenský, J. Vykopal, P. Seda and P. Čeleda, 'Dataset of shell commands used by participants of hands-on cybersecurity training,' *Data in Brief*, vol. 38, p. 107 398, 2021, ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2021.107398. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340921006806.

[54] V. Švábenský, J. Vykopal, P. Čeleda, K. Tkáčik and D. Popovič, 'Student assessment in cybersecurity training automated by pattern mining and clustering,' *Education and Information Technologies*, vol. 27, no. 7, pp. 9231–9262, Aug. 2022, ISSN: 1360-2357. DOI: 10.1007/s10639-022-10954-4. [Online]. Available: https://doi.org/10.1007/s10639-022-10954-4.

[55] P. Lepe, A. Aggarwal, J. Mirkovic, J. Mache, R. Weiss and D. Weinmann, 'Measuring student learning on network testbeds,' in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, 2019, pp. 1–2. DOI: 10.1109/ICNP.2019.8888101.

[56] M. Andreolini, V. G. Colacino, M. Colajanni and M. Marchetti, 'A framework for the evaluation of trainee performance in cyber range exercises,' *Mobile Networks and Applications*, vol. 25, no. 1, pp. 236–247, Feb. 2020, ISSN: 1572-8153. DOI: 10.1007/s11036-019-01442-0. [Online]. Available: https://doi.org/10.1007/s11036-019-01442-0.

[57] C. Zheng, Z. Liu, E. Xie, Z. Li and Y. Li, *Progressive-hint prompting improves reasoning in large language models*, 2023. arXiv: 2304.09797 [cs.CL].

[58] M. Pankiewicz and R. S. Baker, *Large language models (gpt) for automating feedback on programming assignments*, 2023. arXiv: 2307.00150 [cs.HC].

[59] J. McCarthy, 'What is artificial intelligence?,' Jan. 2004.

[60] IBM. (2023), [Online]. Available: https://www.ibm.com/topics/artificial-intelligence (visited on 11/08/2023).

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, 'Attention is all you need,' in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[62] E. Carle. (2023), [Online]. Available: https://blog.google/inside-google/googlers/ask-a-techspert/what-is-generative-ai/ (visited on 08/08/2023).

[63] G. Lawton. (2023), [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/generative-AI (visited on 08/08/2023).

[64] Nvidia. (2023), [Online]. Available: https://www.nvidia.com/en-us/glossary/data-science/generative-ai/ (visited on 08/08/2023).

[65] M. N. Center. (2023), [Online]. Available: https://news.microsoft.com/2023/04/04/ai-explained/ (visited on 11/08/2023).

[66] D. Dodsworth. (2023), [Online]. Available: https://history-computer.com/what-are-large-language-models/ (visited on 11/08/2023).

[67] M. Kwon, S. M. Xie, K. Bullard and D. Sadigh, 'Reward design with language models,' *ArXiv*, vol. abs/2303.00001, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257255456.

[68] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, 'Improving language understanding by generative pre-training,' 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[69] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, 'Language models are few-shot learners,' 2020. DOI: 10.48550/arXiv.2005.14165. [Online]. Available: https://arxiv.org/abs/2005.14165.

[70] R. Child, S. Gray, A. Radford and I. Sutskever, 'Generating long sequences with sparse transformers,' *arXiv preprint arXiv:1904.10509*, 2019.

[71] A. P. IV. 'Major chatgpt update: Ai program no longer restricted to september 2021 knowledge cutoff.' (2023), [Online]. Available: https://www.forbes.com.au/news/innovation/chatgpt-removes-september-2021-knowledge-cutoff/ (visited on 07/10/2023).

[72] A. A. Patel, B. Linton and D. Sostarec. 'Gpt-4, gpt-3, and gpt-3.5 turbo: A review of openai's large language models.' (2023), [Online]. Available: https://www.ankursnewsletter.com/p/gpt-4-gpt-3-and-gpt-35-turbo-a-review (visited on 12/10/2023).

[73] OpenAI. 'Gpt-3.5.' (2023), [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5 (visited on 08/10/2023).

[74] J. Han, M. Kamber and J. Pei, '2 - getting to know your data,' in *Data Mining (Third Edition)*, ser. The Morgan Kaufmann Series in Data Management Systems, J. Han, M. Kamber and J. Pei, Eds., Third Edition, Boston: Morgan Kaufmann, 2012, pp. 39–82, ISBN: 978-0-12-381479-1. DOI: https://doi.org/10.1016/B978-0-12-381479-1.00002-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123814791000022.

[75] V. I. Levenshtein *et al.*, 'Binary codes capable of correcting deletions, insertions, and reversals,' in *Soviet physics doklady*, Soviet Union, vol. 10, 1966, pp. 707–710.

[76] A.-K. Carstensen and J. Bernhard, 'Design science research – a powerful tool for improving methods in engineering education research,' *European Journal of Engineering Education*, vol. 44, no. 1-2, pp. 85–102, 2019. DOI: 10.1080/03043797.2018.1498459. eprint: https://doi.org/10.1080/03043797.2018.1498459. [Online]. Available: https://doi.org/10.1080/03043797.2018.1498459.

[77] F. CSS. 'Free css.' (2023), [Online]. Available: https://www.free-css.com/free-css-templates (visited on 04/07/2023).

[78] L. Statistics. (2023), [Online]. Available: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php (visited on 12/12/2023).

[79] J. Beckman, I. Ferru and E. A. Beckmann, 'Softly, softly: Introducing research-led education into a successful first year course,' in *Proceedings of the Australian Conference of Science and Mathematics Education (2013)*, Sep. 2013.

[80] J. Miller and R. Ulrich, 'The quest for an optimal alpha,' *PLOS ONE*, vol. 14, no. 1, pp. 1–13, Jan. 2019. DOI: 10.1371/journal.pone.0208. [Online]. Available: https://ideas.repec.org/a/plo/pone00/0208631.html.

[81] E.-J. Wagenmakers. 'Jasp - a fresh way to do statistics.' (2018), [Online]. Available: https://jasp-stats.org (visited on 16/10/2023).

[82] J. Love, D. Dropmann, R. Selker, M. Gallucci, S. Jentschke, S. Balci, H. Seol and M. Agosti. 'Jamovi - open statistical software for the desktop and cloud.' (2023), [Online]. Available: https://www.jamovi.org (visited on 15/11/2023).

[83] J. J. 'How can i access gpt-4?' (2023), [Online]. Available: https://help.openai.com/en/articles/7102672-how-can-i-access-gpt-4 (visited on 27/10/2023).

[84] F. Guerrini. 'Ai's unsustainable water use: How tech giants contribute to global water shortages.' (2023), [Online]. Available: https://www.forbes.com/sites/federicoguerrini/2023/04/14/ais-unsustainable-water-use-how-tech-giants-contribute-to-global-water-shortages/ (visited on 28/10/2023).

[85] W. McCurdy. 'How ai tools like chatgpt are causing water usage to skyrocket.' (2023), [Online]. Available: https://www.standard.co.uk/news/tech/ai-chatgpt-water-power-usage-b1106592.html (visited on 28/10/2023).

## Appendix A

# Interview Questions

- Why do you think injection vulnerabilities are still an issue?
- Do you think there is a need to be learning offensive techniques or should we as a community invest more time into defensive security and research?
- How do you find injection vulnerabilities and what is your favorite method/vulnerability?
- Is there any concern/issues in doing offensive security research and/or education?
- Is there a way to stop or at least decrease the likelihood that offensive security education is being used unethically?
- What is the most commonly found injection vulnerability?
- Can you rate the difficulty level of finding and exploiting these injection vulnerabilities:
    - SQL injection
    - Command injection
    - Template injection
    - OS command injection
    - XSS
    - CSRF
    - Formula injection
    - Xpath injection
    - Any other that you can think of? If so, what is its rating?

# Appendix B

# Questionnaires

## B.1 The Norwegian Defence Cyber Academy

### B.1.1 Hint group

Nettskjema

# User experience of Injection lab

## The following questions are about the lab itself.

### How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging)

1
2
3
4
5

### Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree)

1
2
3
4
5

### How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly)

1
2
3
4
5

## Do you have any comments or feedback on the lab itself?

## The following questions is about you experienced the hints that you were given during the lab exercises.

### Were the frequency and quantity of hints sufficient to assist in your progress with the challenges? (1: Strongly disagree, 5: Strongly agree)

1
2
3
4
5

### Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree)

1
2
3

4

5

### Did the content of the hints facilitate steady progress with the challenges without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree)

1

2

3

4

5

### Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)

1

2

3

4

5

## Do you have any comments or feedback to the provided hints?

## The following questions is about the learning material that you were provided in the lab.

On a scale 1 to 5, how much do you agree with the following statements.
Where 1 being strongly disagree, and 5 being strongly agree.

### How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear)

1

2

3

4

5

### To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful)

1

2

3

4

5

### Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful)

1

2

3

4

5

**Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent)**

1

2

3

4

5

### Do you have any comment or feedback on the learning material?

### Is there any additional comments or feedback about the project that you would like to provide?

### B.1.2 Control group

# User experience of injection lab

## The following questions are about the lab itself.

Please rate the following statements.

**How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging)**

1
2
3
4
5

**Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly)**

1
2
3
4
5

## Do you have any comments or feedback on the lab itself?

## The following questions is about how you experienced the challenges you were given during the lab.

What was challenging in the exercises?

Please rate the following statements.

**Was the exercises challenging? (1: Not at all, 5: Exremely challenging)**

1
2
3
4
5

**Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much)**

1

2

3

4

5

**Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful)**

1

2

3

4

5

## Do you have any comments or feedback to the challenges?

## The following questions is about the learning material that you were provided in the lab.

Please rate the following statements.

**How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear)**

1

2

3

4

5

**To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful)**

1

2

3

4

5

**Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful)**

1

2

3

4

5

**Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent)**

1

2

3

4

5

## Do you have any comment or feedback on the learning material?

## Is there any additional comments or feedback about the project that you would like to provide?

## B.2 NTNU

### B.2.1 Hint group

# User experience of Injection lab

## The following questions is about the lab itself.

Please rate the following statements.

**How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging)**

1
2
3
4
5

**Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly)**

1
2
3
4
5

## Do you have any comments or feedback on the lab itself?

## The following questions is about how you experienced the hints that you were given during the lab exercises.

Please rate the following statements.

**Were the frequency and quantity of hints sufficient to assist in your progress with the challenges? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did the content of the hints facilitate steady progress with the challenges without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

## Do you have any comments or feedback to the provided hints?

## The following questions is about how you experienced the challenges you were given during the lab.

What was challenging in the exercises?

Please rate the following statements.

**Was the exercises challenging? (1: Not at all, 5: Exremely challenging)**

1

2

3

4

5

**Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2
3
4
5

**Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much)**

1
2
3
4
5

**Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful)**

1
2
3
4
5

## Do you have any comments or feedback to the challenges?

## The following questions is about the learning material that you were provided in the lab.

Please rate the following statements.

**How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear)**

1
2
3
4
5

**To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful)**

1
2
3
4
5

**Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful)**

1
2
3

4

5

**Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent)**

1

2

3

4

5

### Do you have any comment or feedback on the learning material?

### Is there any additional comments or feedback about the project that you would like to provide?

### B.2.2 Control group

Nettskjema

# User experience of injection lab kopi

## The following questions are about the lab itself.

Please rate the following statements.

**How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging)**

1
2
3
4
5

**Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly)**

1
2
3
4
5

## Do you have any comments or feedback on the lab itself?

## The following questions is about how you experienced the challenges you were given during the lab.

What was challenging in the exercises?

Please rate the following statements.

**Was the exercises challenging? (1: Not at all, 5: Exremely challenging)**

1
2
3
4
5

**Did you make steady progress through the challenges?? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much)**

1

2

3

4

5

**Would hints have been helpful during the exercises?(1: Not at all, 5: Extremely helpful)**

1

2

3

4

5

## Do you have any comments or feedback to the challenges?

## The following questions is about the learning material that you were provided in the lab.

Please rate the following statements.

**How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear)**

1

2

3

4

5

**To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful)**

1

2

3

4

5

**Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful)**

1

2

3

4

5

**Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent)**

1

2

3

4

5

## Do you have any comment or feedback on the learning material?

## Is there any additional comments or feedback about the project that you would like to provide?

## B.3 CTF

Nettskjema

# User experience of Template Injection challenge

## The following questions is about the lab itself.

Please rate the following statements.

**How do you assess the level of difficulty posed by the lab task? (1: Not challenging at all, 5: Extremely challenging)**

1
2
3
4
5

**How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly)**

1
2
3
4
5

## Do you have any comments or feedback on the lab itself?

## The following questions is about how you experienced the hints that you were given during the challenge.

Please rate the following statements.

**Were the frequency and quantity of hints sufficient to assist in your progress with the challenge? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree)**

1
2
3
4
5

**Did the content of the hints facilitate steady progress with the challenge without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

## Do you have any comments or feedback to the provided hints?

## The following questions is about how you experienced the challenge.

What was challenging?

Please rate the following statements.

**Was it challenging? (1: Not at all, 5: Exremely challenging)**

1

2

3

4

5

**Did you make steady progress through the challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

**Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree)**

1

2

3

4

5

## Do you have any comments or feedback to the challenge?

## Is there any additional comments or feedback about the project that you would like to provide?

# Appendix C

# Surveys

Nettskjema

# Pre-Survey

### Question 1: What is your experience with offensive security and template injection vulnerabilities?

None

Basic understanding

Intermediate knowledge

Advanced knowledge

### Question 2: How confident are you in your ability to detect injection vulnerabilities?

Not confident at all

Slightly confident

Moderatley confident

Very confident

### Question 3: What is the cause of template injection vulnerabilities?

A lack of encryption

A lack of proper input validation

Missing security software

A lack of log monitoring

### Question 4: What are some potential consequences of a successful injection attack on web applications?

Data leakage

Unauthorized access

Code execution

None of the above

### Question 5: What are some common security threats or attack that web applications may face?

Cross-Site Scripting (XSS)

Cross-Site Request Forgery (CSRF)

SQL Injection

None of the above

### Question 6: What steps can be taken to ensure that data provided by users is safe when interacting with a web application?

I have no idea

Enable firewall rules to restrict network traffic

Implement error handling

Encrypt the data before processing

Employ data examination processes

### Question 7: What are some common types of data that requires input validation in

### web applications?

Username and passwords

Credit card numbers

Email addresses

None of the above

### Question 8: Which of the following best describes the role of input validation in web application security and its effects on data integrity, confidentiality, and availability?

Input validation ensures data confidentiality by encrypting user inputs, making them unreadable to unauthorized parties

Input validation verifies the integrity of user inputs and protects data from unauthorized changes, maintaining its accuracy and consistency

Input validation enhances system availability by optimizing server resources and ensuring fast response times

Input validation secures data confidentiality and availability by blocking all user inputs to prevent any potential threats

### Question 9: Which of the following is a temporary measure to mitigate the risk of injection flaws in a web application while a permanent solution is being developed?

Applying strong encryption to all data in the database

Implementing a web application firewall (WAF)

Implement encryption of user input before processing

Configuring an intrusion detection system (IDS)

### Question 10: What is a common mitigation technique(s) to prevent injection vulnerabilities in web applications?

Disable user input

Blacklisting special characters

Input validation and filtering

Running security scans regularly

Nettskjema

# Post-survey

### What username did you use?

### How would you rate your understanding of template injection vulnerabilities?

Poor

Fair

Good

Excellent

### How would you mitigate a template injection vulnerability in a web application?

Regulary monitor the server logs

Implement input validation and output encoding

Use strong encryption for data transmission

Install antivirus software

Update the software the application uses

### What are some  common security threats or attack that web applications may face?

Cross-Site Scripting (XSS)

Cross-Site Request Forgery (CSRF)

SQL Injection

Server-Side Template Injection (SSTI)

None of the above

### What can a successful template injection attack on a web server lead to?

Information disclosure

Unauthorized access

Code execution

None of the above

### What are some common types of input fields which requires input validation in a web application?

Username and passwords

Search fields

Email addresses

None of the above

### What is the most effective mitigation against injection vulnerabilities?

Running regularly security scans

Input validation and filtering

Disable user input entirely

Increasing server response times

### What can be done to avoid template injection vulnerabilities?

Apply secure coding practices and avoid using templates

Use secure functions for rendering web pages and sanitize user input
Encode payloads with base64 or similar encodings
Use publicly available templates without modifications

## Is it necessary to review default settings in web applications?

Default settings are always secure and require no modification
Modifying default settings improves user experience
Default settings may expose vulnerabilities and should be customized
Default settings are automatically adjusted by the server, therefore there is no need to adjust them yourself

## What function does input validation serve?

It ensures that user input is encrypted before processing
It ensures that user input matches a expected pattern before processing
It ensures that the input is stored safely for later processing
I have no idea

## How does effective input validation in a web application contribute to maintaining the principles of confidentiality, integrity, and availability of data and resources?

Input validation primarily ensures data integrity by preventing unauthorized changes to information
Input validation enhances data confidentiality by encrypting user inputs to safeguard against unauthorized access
Input validation has no significant impact on system availability as it focuses on data integrity and confidentiality
Input validation supports data integrity, confidentiality, and availability by preventing unauthorized access and manipulation

# Appendix D

# Consent form

# Vil du delta i forskningsprosjektet

# *Automatisk hint generering for Injection lab*?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke hvordan autogenererte hint kan bidra i læringen av injection sårbarheter. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

**Formål**

*Formålet med prosjektet er å effektivisere tilegningen av kunnskap relatert til injection sårbarheter, hvordan de oppstår, hvordan man kan finne dem og hvordan fikse dem. Prosjektet vil ta for seg et få tall typer injection sårbarheter for å fokusere på undervisningsdelen av prosjektet.*
*Vi ønsker å undersøke hvor effektiv hint som er autogenerert av tidligere løsningsforslag og som er tilpasset deltakerens fremgangsmåte kan bidra i læringsprosessen av injection sårbarheter.*
*Dette prosjektet er en masteroppgave.*

**Hvem er ansvarlig for forskningsprosjektet?**

*NTNU Gjøvik* er ansvarlig for prosjektet.

**Hvorfor får du spørsmål om å delta?**

*Du får spørsmål om å delta fordi du går en teknisk utdanning med fag innenfor informasjonssikkerhet.*

**Hva innebærer det for deg å delta?**

Hvis du velger å delta i prosjektet, innebærer det at du prøver å fullføre flere laboppgaver som er designet for å lære deg hvorfor injection sårbarheter oppstår og hvordan finne noen typer.
Etter laboppgavene fyller du et digitalt spørreskjema om hvordan du opplevde oppgavene. Det vil ta ca. 5 minutter. Svarene dine vil bli brukt til å forbedre labene og vurdere hvor effektiv prosjektet er i læring av injection sårbarheter.

Det vil bli samlet anonym data om hvordan du integrerer med labbene. Dataene vil også bli benyttet til å analysere effektiviteten og muligens identifisere svakheter med prosjektet.

**Det er frivillig å delta**

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Deltakelse i forskningen inngår ikke i normal undervisning og er derfor ikke grunnlag for vurdering av deltakerne.

**Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Dataen vil være tilgjengelig for veilederne og studenten for dette prosjektet. Dataen vil være lagret i cyber rangen til NTNU hvor kun autorisert personell vil ha tilgang.

Data som blir samlet inn fra deg vil være anonymisert og aldri være knyttet til informasjon som kan knyttet til det som individ.

Dataen vil samles inn av systemet som er satt opp i cyber rangen til NTNU. Dataene vil bli bearbeidet av studenten og veilederne.

**Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?**
Prosjektet vil etter planen avsluttes 15. desember 2023. Da prosjektet ikke samler inn personopplysninger og all data er anonymisert vil ikke data bli slettet, og kan bli gjenbrukt til eksempelvis forskning.

**Hva gir oss rett til å behandle personopplysninger om deg?**
Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra NTNU har Sikt – Kunnskapssektorens tjenesteleverandør vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

**Dine rettigheter**

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

- NTNU Gjøvik ved Basel Katt
  E-post: basel.katt@ntnu.no
  Mobil: 61135176

  John Martin Johnsen
  E-post: JM_Nesna@hotmail.com
  Mobil: 48228321
- Vårt personvernombud: Thomas Helgesen
  E-post: thomas.helgesen@ntnu.no
  Mobil: 93079038

Hvis du har spørsmål knyttet til vurderingen som er gjort av personverntjenestene fra Sikt, kan du ta kontakt via:

- Epost: personverntjenester@sikt.no eller telefon: 73 98 40 40.

Med vennlig hilsen

*John Martin Johnsen*

Masterstudent

------------------------------------------------------------------------------------------------------------------------

# Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet "Automatisk hint generering for Injection lab", og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i gjennomføring av injection lab
- å delta i spørreskjema

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---------------------------------------------------------------------------------------------------------------

(Signert av prosjektdeltaker, dato)

# Appendix E

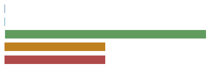
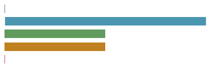
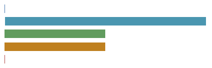
# Questionnaire responses

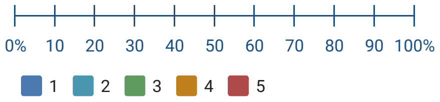
## E.1 NTNU

# User experience of Injection lab

Oppdatert: 9. desember 2023 kl. 10:58

## The following questions is about the lab itself.

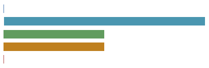
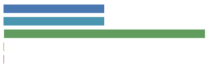
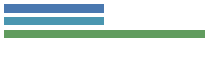
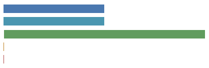
| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | | | 2 | 1 | 1 |  |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | | 2 | 1 | 1 | |  |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | | 2 | 1 | 1 | |  |

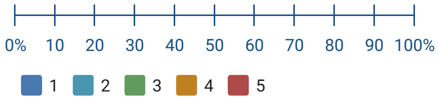


## Do you have any comments or feedback on the lab itself?

Dette spørsmålet har ingen svar

## The following questions is about how you experienced the hints that you were given during the lab exercises.

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| Were the frequency and quantity of hints sufficient to assist in your progress with the challenges? (1: Strongly disagree, 5: Strongly agree) |  | 2 | 1 | 1 |  |  |
| Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree) | 1 | 1 | 2 |  |  |  |
| Did the content of the hints facilitate steady progress with the challenges without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree) | 1 | 1 | 2 |  |  |  |
| Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | 1 | 1 | 2 |  |  |  |



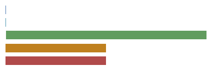
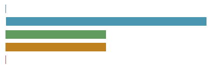
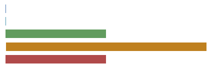
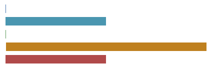
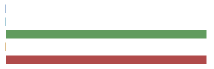
## Do you have any comments or feedback to the provided hints?

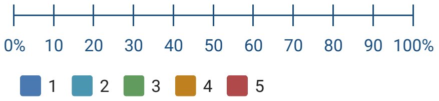
Dette spørsmålet har ingen svar

## The following questions is about how you experienced the challenges you were given during the lab.

- Figuring out how to start the first task, rest was ok

## Spørsmål uten tekst

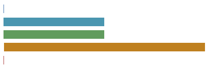
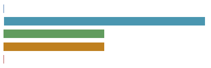
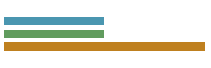
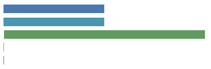
| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| Was the exercises challenging ? (1: Not at all, 5: Exremely challenging ) |  |  | 2 | 1 | 1 |  |
| Did you make steady progress through the challenges ?? (1: Strongly disagree , 5: Strongly agree) |  | 2 | 1 | 1 |  |  |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge ? (1: Strongly disagree , 5: Strongly agree) |  |  | 1 | 2 | 1 |  |
| Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much) |  | 1 |  | 2 | 1 |  |
| Would hints have been helpful during the exercises ? (1: Not at all, 5: Extremely helpful ) |  |  | 2 |  | 2 |  |

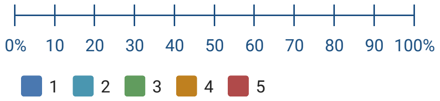


## Do you have any comments or feedback to the challenges?

Dette spørsmålet har ingen svar

## The following questions is about the learning material that you were provided in the lab.

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) |  | 1 | 1 | 2 |  |  |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) |  | 2 | 1 | 1 |  |  |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) |  | 1 | 1 | 2 |  |  |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) | 1 | 1 | 2 |  |  |  |



## Do you have any comment or feedback on the learning material?

Dette spørsmålet har ingen svar

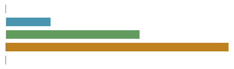
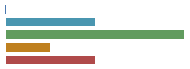
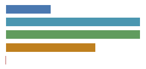
## Is there any additional comments or feedback about the project that you would like to provide?

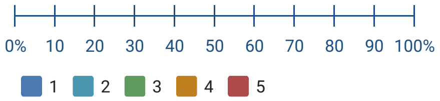
Dette spørsmålet har ingen svar

# User experience of injection lab kopi

Oppdatert: 9. desember 2023 kl. 10:59

## The following questions are about the lab itself.

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| How do you assess the level of difficulty posed by the lab tasks? (1: Not challenging at all, 5: Extremely challenging) | | 1 | 3 | 5 | |  |
| Did the sequence of challenges in the lab facilitate a deeper understanding of the vulnerability under investigation? (1: Strongly disagree, 5: Strongly agree) | | 2 | 4 | 1 | 2 |  |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 1 | 3 | 3 | 2 | |  |



## Do you have any comments or feedback on the lab itself?

- No comment

- Got random Hints, including the answer for step 4 at step 1. I made alot of attemptsI misunderstood the order i was supposed to read through for testing for the engine and got stuck thinking i was working on Expression Language.The hints also straight up gave me the answer for step 4, and i really didnt understand the concept of the syntax.

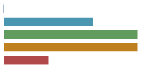
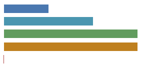
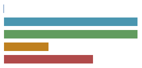
- The lab itself was a normal lab. Nothing specially bad, nor specially good.

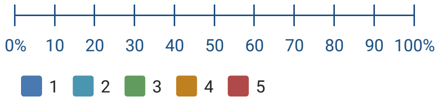
- Learning by practice is what works best for me, so I found it effective. There were some rough edges though, for example I passed task 3 by inputting only one of 7 blacklisted characters (`#`), and the questionnaires had issues loading and submitting.An easy thing to fix: the bottom field to submit the solution to the exercise has its placeholder text (the question to the field) set to a colour that is very difficult to see against the white field, on my old laptop's TN panel I initially didn't notice the text at all.

## The following questions is about how you experienced the challenges you were given during the lab.

- Finding correct input for task 4. I heard those that got hints got a working payload presented to them. Those that didn't had to read a bit, understand some and modify and copy a payload from the provided git-repo.
- Understanding how to read serverside code with inspect tool.Learning to code in another language on the fly (or understanding syntax in a language I'm not familiar with).
- Finding the right command to use to identify the correct text-field, and also actually finding all the text-fields to try (the last one is on me though). Also I ran into an issue where it seemed like there are different engines used when it should be the same
- There was so many alternatives to try. This is pretty exhausting and boring. And you don't really get any value out of that. After a while it gets frustrating when nothing works.
- the last challenge without hints ({{""}}, {{"}}, {{'}}, {{"}}, {{()}}, {{[]}} did not work)

## Spørsmål uten tekst

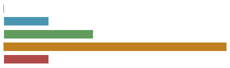
| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| Was the exercises challenging ? (1: Not at all, 5: Exremely challenging ) |  | 2 | 3 | 3 | 1 |  |
| Did you make steady progress through the challenges ?? (1: Strongly disagree , 5: Strongly agree) | 1 | 2 | 3 | 3 |  |  |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge ? (1: Strongly disagree , 5: Strongly agree) |  | 3 | 3 | 1 | 2 |  |
| Did you find the exercises reflected the learning material well? (1: Not at all, 5: Very much) |  | 1 | 2 | 6 |  | |
| Would hints have been helpful during the exercises ? (1: Not at all, 5: Extremely helpful ) | 1 |  | 1 | 3 | 4 | |

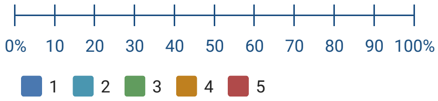


## Do you have any comments or feedback to the challenges?

- Not more then what I already wrote.
- relevant, good :)

## The following questions is about the learning material that you were provided in the lab.

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| How would you rate the clarity of the explanations provided in the learning material? (1: Very unclear, 5: Very clear) |  | 2 | 2 | 4 | 1 | |
| To what extent did the provided examples contribute to your comprehension of the concepts covered in the learning material? (1: Not helpful at all, 5: Extremely helpful) |  |  | 3 | 6 |  | |
| Did you find the organization and flow of the learning material to be logical and easy to follow? (1: Not at all, 5: Extremely helpful) |  | 1 | 2 | 5 | 1 |  |
| Please evaluate the quality of the visual presentation, including formatting and graphics, in the learning material. (1: Poor, 5: Excellent) |  |  | 4 | 4 | 1 | |



## Do you have any comment or feedback on the learning material?

- The hints were helpful. I feel like i didnt have a good foundation for doing the test so it was a very new experience. Maybe send out some preparation material ( about engines for example) in advance? i feel like this was a useful experience that will benefit me going forward.
- The different steps in the learning material could have been more clearly separated.
- The learning material was quite good. But it was still challenging when it came time to actually do the exercise.

## Is there any additional comments or feedback about the project that you would like to provide?

- The Wifi here is slow at times, so when submitting many times to check for injection points, it took quite a bit of time. (not sure this is something you can do anything about)
- The learning material was general, and that is not so helpful when it comes time to actually do the exercise. Customized hint would definitely have been helpful.

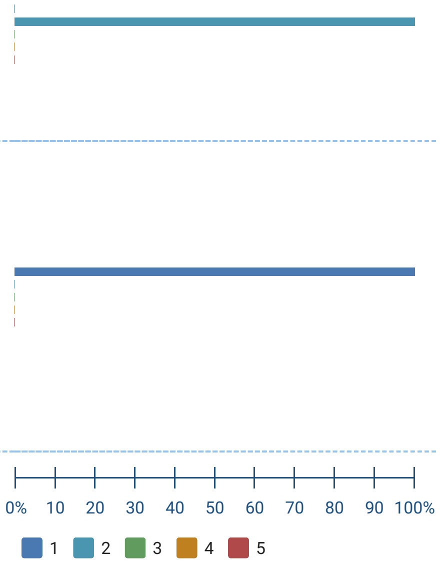
## E.2 CTF

# User experience of Template Injection challenge

Oppdatert: 9. desember 2023 kl. 11:00

## The following questions is about the lab itself.

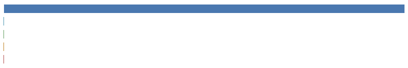
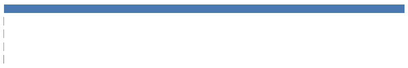
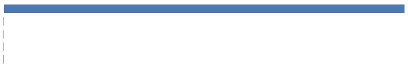
| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| How do you assess the level of difficulty posed by the lab task? (1: Not challenging at all, 5: Extremely challenging) | | 1 | | | | |
| How would you rate the user-friendliness of the lab interface concerning ease of navigation and prevention of confusion? (1: Very confusing, 5: Very user-friendly) | 1 | | | | | |

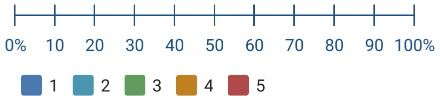


## Do you have any comments or feedback on the lab itself?

- I have many things I want to complain about:1. The application gave bad hints! It sometimes talked about JavaScript and SQL injections, which had nothing to do with this task. This was super confusing and unnecessary.2. It was not clear where we should input the template injection payload. There was way to many input fields, and when the hints talked about the "advanced" filters we had to bypass it lead me to think that the input fields I tried had STI vulns, but I had the wrong payload.3. It was super easy to do the injection when I found the right input field. Like... there was no filtering at all. Using a standard injection I got the flag in an instant.

## The following questions is about how you experienced the hints that you were given during the challenge.

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| Were the frequency and quantity of hints sufficient to assist in your progress with the challenge? (1: Strongly disagree, 5: Strongly agree) | 1 | | | | |  |
| Were the hints applicable to the specific aspects you were attempting to manipulate or exploit? (1: Strongly disagree, 5: Strongly agree) | 1 | | | | |  |
| Did the content of the hints facilitate steady progress with the challenge without leading to complete impasses? (1: Strongly disagree, 5: Strongly agree) | 1 | | | | |  |
| Did the hints diminish the sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge? (1: Strongly disagree, 5: Strongly agree) | | 1 | | | | |



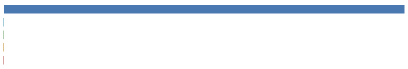
## Do you have any comments or feedback to the provided hints?

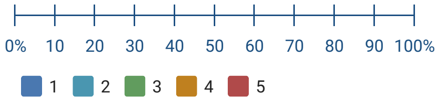
- The hits was bad... They talked about things that was totally unrelevant to the solution of the challenge. They should instead pointed the user to the correct input field and tell us that the application is running python. NOT JavaScript as some of the hints would like us to think.

## The following questions is about how you experienced the challenge.

- No

## Spørsmål uten tekst

| Svar | 1 | 2 | 3 | 4 | 5 | Diagram |
|---|---|---|---|---|---|---|
| Was it challenging ? (1: Not at all, 5: Exremely challenging ) | 1 | | | | | |
| Did you make steady progress through the challenge ? (1: Strongly disagree , 5: Strongly agree) | 1 | | | | |  |
| Did you feel a sense of satisfaction and accomplishment you typically experience upon successfully completing a challenge ? (1: Strongly disagree , 5: Strongly agree) | 1 | | | | | |



## Do you have any comments or feedback to the challenge?

- Give us the source code. This way it would be more fun :))

## Is there any additional comments or feedback about the project that you would like to provide?

- Don't include these types of challenges in S2G. This challenge is was not a typical S2G challenge

NTNU

Norwegian University of
Science and Technology