

Neon City Defender: When A Serious Game Meets Web Application Security Teaching

Wenjia Geng



Master of Science
School of Informatics
University of Edinburgh
2024

Abstract

With the rapid development of the internet, web applications have become deeply embedded in daily life, while cybercrime has also increased in prevalence. This has exacerbated the global shortage of cybersecurity talent. File upload vulnerabilities are common yet often overlooked high-risk vulnerabilities in web applications.

To address this issue and familiarize students with the threats and defences associated with file upload vulnerabilities, this study designed and developed a serious game called Neon City Defender. The game aims to teach undergraduate students the identification, exploitation, and remediation of file upload vulnerabilities in web application security. By introducing CTF Jeopardy-style challenges, the game enables students to engage in practice under simulated real-world scenarios, closing the divide between theoretical understanding and practical implementation. Additionally, the serious game addresses common issues in CTF-based teaching, including high prior knowledge barriers that undermine students' confidence and a lack of sustained motivation for students.

Neon City Defender is implemented upon the LM-GM serious game design model [35] and utilizes an objective-driven design methodology proposed in the research. It used students' learning and game motives in the context of instructional objectives that matched the learning mechanics (LMs) and gaming mechanics (GMs) of the LM-GM model. Leveraging these LMs and GMs, the game implements ten core serious game mechanics (SGMs) across task, reward, and feedback aspects, ensuring a balance between educational functionality and entertainment while effectively meeting its instructional objectives. Evaluations of Neon City Defender's functionality, usability, and teaching effectiveness demonstrate that all features perform as intended, with strong usability and significant educational impact.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.

Ethics application number: 945616

Date when approval was obtained: 2024-06-07

The participants' information sheet and a consent form are included in the appendix.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Wenjia Geng)

Acknowledgements

As I approach the completion of my dissertation, I want to extend my heartfelt gratitude to my supervisor, teachers, family, and friends for their unwavering support, guidance, and encouragement during my year as a graduate student.

Firstly, I am profoundly grateful to my supervisor, Dr. Myrto D. Arapinis, whose invaluable academic mentorship and ongoing assistance during the research process helped me navigate numerous obstacles.

I am also sincerely thankful to the students who took part in the user research. Their contributions provided comprehensive and valuable experimental data, forming a solid foundation for the evaluation of the Neon City Defender.

I would like to thank my family for their unwavering support and encouragement, which allowed me to focus on my studies and complete my dissertation successfully.

With this deep sense of gratitude and hope for the future, I will continue to move forward, transforming this appreciation into the motivation and strength needed to pursue future achievements.

Table of Contents

1	Introduction	1
1.1	Project Aim	2
1.2	Contribution	3
1.3	Report Structure	4
2	Background	5
2.1	File Upload Vulnerabilities	5
2.2	CTF-Based Cybersecurity Education	6
2.3	Serious Games	6
2.3.1	The Concept of Serious Game	6
2.3.2	Application of Serious Games in Education	7
2.3.3	Serious Game Development Framework	7
2.3.4	Serious Game Design Model	7
2.4	Summary	8
3	Requirements Gathering and Analysis	9
3.1	Objective-Driven Design Methodology	9
3.2	Survey on Learning and Game Motives	10
3.2.1	Questionnaire Results Analysis	11
3.3	Case Study on LMs and Game GMs	11
3.3.1	Case Study of Web Security Education Platforms	12
3.3.2	Case Study of Cybersecurity Serious Games	13
3.4	Functional Requirements and Use Cases	15
3.5	Summary	15
4	Design	16
4.1	Serious Game Mechanics	16
4.1.1	Task Mechanics	17

4.1.2	Reward Mechanics	18
4.1.3	Feedback Mechanics	19
4.2	Gameplay Flow	20
4.3	Game Background Story	21
4.4	Game Levels	22
4.4.1	Level Challenge Page Design	22
4.4.2	Level Design	23
4.5	UI Design	24
4.6	Summary	24
5	Implementation	25
5.1	System Architecture of the Serious Game	25
5.2	Cloud Deployment	26
5.3	Serious Game User Interface	27
5.4	Core Technologies Applied	27
5.5	Implementation of Core SGMs	29
5.5.1	Subtask Completion Recognition	29
5.5.2	Vulnerability Remediation Practice	30
5.6	Summary	31
6	Evaluation	32
6.1	Functionality Evaluation	32
6.2	Evaluations Involving User Research	32
6.2.1	Participants	33
6.2.2	User Research Process	33
6.3	Usability Evaluation	33
6.3.1	Semi-Structured Interviews	34
6.3.2	SUS Questionnaire	36
6.4	Teaching Effectiveness Evaluation	37
6.5	Summary	38
7	Conclusion	39
7.1	Overview	39
7.2	Further Work	40
	Bibliography	41

A	Essential Supporting Materials	47
A.1	The LM-GM Serious Games Design Model	47
A.2	Serious Game System Functional Requirements	48
A.3	UML Diagrams for Serious Game Use Cases	50
A.4	UI Optimizations using Nielsen’s Ten Heuristics	51
A.5	Communication in the Serious Game System	52
A.5.1	Client-to-Server Requests (Green Arrow)	52
A.5.2	Database Communication (Purple Arrow)	53
A.5.3	Docker API Communication (Blue Arrow)	53
A.5.4	Internal VM Network Communication (Red Arrow)	54
A.6	Code Snippets of Core SGMs Implementation	55
A.7	System Usability Scale Evaluation Standards	56
B	Questionnaire for Collecting Learning and Game Motives	57
C	System Functional Test Cases	60
D	Semi-structured Interview Questions	65
E	SUS Questionnaire	66
F	Pre-and Post-tests Questions for Teaching Effectiveness Evaluation	68
G	Gameplay UI Implementation Screenshots	71
H	Participants’ information sheet	81
I	Participants’ consent form	86

Chapter 1

Introduction

With the rapid development of the Internet, the widespread adoption of web applications has accelerated enterprises' transition to online business [33]. Web applications have become deeply integrated into people's daily lives [46]. However, while many applications enhance user experience, they often neglect security. Since 2020, web application vulnerabilities have become a common and serious threat in the field of cybersecurity [26]. Data shows that approximately 91% of web applications have experienced sensitive information leakage due to security vulnerabilities [1], leading to a surge in demand for web application security professionals by enterprises. In 2023, the global cybersecurity talent gap reached 3.5 million, and this supply-demand imbalance is expected to persist until 2025 [39]. Therefore, training cybersecurity professionals who possess the skills to identify, exploit, and remediate web application vulnerabilities has become a top priority.

File upload vulnerabilities are among the most common and highly threatening vulnerabilities in web applications. According to CVE reports, file upload-related vulnerabilities are often rated as high-risk or critical, and attackers can exploit these vulnerabilities to execute arbitrary code on affected servers [20][21][22]. Furthermore, file upload functionality is prevalent in web applications, ranging from profile avatar uploads to data backups. Once a malicious file (such as a web shell) is successfully uploaded and executed, attackers can fully control the server without bypassing other security measures [44], potentially leading to sensitive data leaks, service interruptions, and even using the compromised server as a springboard for further attacks. Therefore, teaching knowledge related to file upload vulnerabilities has become a crucial part of web security education, laying the foundation for future vulnerability discovery and remediation.

CTF (Capture The Flag) is a competition format where participants solve cybersecurity challenges to capture "flags" and earn points. CTF not only encourages participants to learn cybersecurity knowledge independently but also provides them with practical experience [47]. Many universities utilize CTF competitions to help students combine cybersecurity theory with practice, deepening their understanding of the knowledge [54]. However, CTF as a teaching tool also has issues, such as high entry barriers and a lack of continuous motivation [16]. Firstly, CTFs usually require knowledge and experience beyond the scope of university courses. Secondly, the high difficulty makes it hard for beginners to quickly get involved in the competition. When faced with complex challenges, they may lose confidence if they are unable to complete tasks, and may even give up in the end.

Serious games, a type of functional game, are widely used in cybersecurity education. By simulating real-world cybersecurity incidents and processes, they make teaching more engaging [14]. Through interaction with game elements, students can safely experiment with and validate cybersecurity techniques in a virtual environment, and gaining knowledge. Combining dull teaching content with engaging game mechanics can effectively increase students' participation and enthusiasm. Additionally, the progressive task difficulty design in serious games provides beginners with a more accessible learning experience, allowing them to quickly complete tasks and receive positive feedback. As the game progresses, students have the opportunity to delve deeper and master more cybersecurity knowledge and skills [48].

1.1 Project Aim

The aim of this project is to design and implement a CTF-style serious game as a teaching tool for teaching students on the identification, exploitation, and remediation of file upload vulnerabilities in web application security. This tool will be designed and developed for undergraduate students with some knowledge of programming or computer security. It will help students who play this serious game understand the causes and principles behind file upload vulnerabilities and acquire the relevant knowledge and skills for exploiting and remediating these vulnerabilities.

By using "capture the flag" approach, students will have the chance to engage in hands-on learning in simulated environments, thus eliminating the disconnect between theory and practical application. The serious game format addresses the challenges of high knowledge barriers and lack of continuous motivation that are often encountered in

CTF competitions. This approach is intended to ignite students' enthusiasm for learning web security and attract more students to join the field of cybersecurity studies. To achieve the above goal, the project will accomplish the following objectives:

1. Enable students to learn and practice the knowledge and skills related to identifying, exploiting, and remediating file upload vulnerabilities through hands-on exercises in the game.
2. Enhance student engagement in completing game tasks through in-game motivation mechanics, encouraging students to explore and learn more about file upload vulnerabilities.
3. Allow students to quickly immerse themselves in the game during the early stages through progressive difficulty design, and build confidence by receiving continuous positive feedback upon task completion.

To verify the effectiveness of using a serious game as a teaching tool for teaching students on file upload vulnerabilities, the project will evaluate its functionality, usability, and teaching effectiveness. Functionality evaluation will be based on functional testing; usability evaluation will be conducted through semi-structured interviews and the System Usability Scale (SUS) questionnaire; and teaching effectiveness will be assessed through pre-and post-tests of the students before and after the serious game.

1.2 Contribution

This study designed and implemented a serious game for teaching the knowledge of identifying, exploiting, and remediating file upload vulnerabilities in web application security. The study proposed an objective-driven design methodology based on the Learning Mechanics-Game Mechanics (LM-GM) serious game design model [35], addressing the issue of effectively relate serious game mechanics (SGMs) with instructional objectives, which provides valuable reference for applying the LM-GM model in the development of serious games for web application security education.

For the first time, this study introduced a mechanics for learning vulnerability fixing through practical code modification within a serious game, innovatively teaching knowledge related to the defence against file upload vulnerabilities. Finally, the study validated the effectiveness of serious games in addressing the high knowledge barriers and lack of sustained motivation associated with the application of CTF in teaching and learning.

1.3 Report Structure

Chapter 2 - Background introduces web application security, file upload vulnerabilities, CTF-based web security education, and the theoretical foundation for serious game development.

Chapter 3 - Requirements Gathering and Analysis clarifies the learning and game motive requirements for the serious game through user research and case studies, refining the system's functional requirements and use cases.

Chapter 4 - Design outlines the design of serious game mechanisms (SGMs), gameplay flow, story background, levels, and UI.

Chapter 5 - Implementation details the system architecture, user interface, key technologies, and the implementation of core SGMs.

Chapter 6 - Evaluation evaluates the serious game's functionality, usability, and teaching effectiveness.

Chapter 7 - Conclusion summarizes the project's outcomes, design, implementation, and evaluation processes, and provides suggestions for future improvements.

Chapter 2

Background

2.1 File Upload Vulnerabilities

Among the many backend vulnerabilities in web applications, file upload vulnerabilities are often overlooked but can pose serious risks. Attackers can exploit such vulnerabilities to upload files containing malicious code or scripts to the server. Once these malicious files are interpreted and executed on the server side, they may cause the web application to malfunction, or even lead to a series of malicious activities such as complete takeover of the service, session hijacking, data destruction, and more [44].

File upload vulnerabilities can be roughly divided into the following three categories: invalid file type validation, invalid file content validation, and improper configuration of file storage location and permissions. Based on this classification, the exploitation of file upload vulnerabilities can be divided into two main types: bypassing frontend validation and bypassing backend validation. The latter is more commonly used by attackers and includes several key techniques: 1) File extension bypass: changing the file extension of the uploaded file to a legitimate extension (e.g., “.gif”); 2) File signature bypass: adding a legitimate file header signature (e.g., the PNG signature “89 50 4E 47 0D 0A 1A 0A”) to the malicious file; 3) Content-Type bypass: modifying the Content-Type field in the upload request to a legitimate type (e.g., “image/jpeg”);

Common defences against file upload vulnerabilities, in addition to the two methods of setting the upload directory as non-executable, randomly renaming file names and paths, the most effective method is to validate file extensions and file content based on whitelist or blacklist rules. File content validation is typically achieved by checking the Content-Type header and file signature.

2.2 CTF-Based Cybersecurity Education

CTF (Capture The Flag) competitions originated at the 1996 DEFCON global hacker conference and are a form of technical competition among cybersecurity professionals in a controlled environment. CTF competitions are mainly divided into three modes: Jeopardy, Attack-Defence, and Mixed [19]. In Jeopardy mode, participants act as attackers, solving challenges related to cryptography, web penetration, reverse engineering, etc., to capture flags and earn points [52].

With the increasing demand for cybersecurity professionals, CTF competitions have been widely used in cybersecurity training to enhance students' practical skills and learning enthusiasm [4]. However, challenges remain in actual teaching. Mirkovic et al. [37] organized Class Capture The Flag (CCTF) exercises through short-term training, which improved some students' cybersecurity skills and interest, but students with weaker foundations often lost confidence due to the wide skill gap and withdrew from the competition, making it difficult for them to effectively participate and benefit. On the other hand, Ford et al. [28] proposed the CTF Unplugged project, which teaches cybersecurity knowledge to students with no background. Although the project improved students' knowledge and confidence, the challenges were not well-aligned with classroom content and lacked an evaluation mechanism, failing to continuously motivate students to learn.

2.3 Serious Games

2.3.1 The Concept of Serious Game

The concept of serious games was introduced by Clark Abt in the 1970s in his book *Serious Game* [2]. He argued that serious games should have explicit and well-considered instructional objectives, possessing both entertainment and functional attributes. In a narrow sense, serious games are primarily designed for functional purposes such as education, therapy, training, and cultural dissemination, while retaining the entertainment aspect throughout the design and development process. However, defining serious games solely from the perspective of functionality and playability has its limitations. Djaouti et al. [24] pointed out that traditional entertainment games can also enhance their serious dimensions through goal shifts and adaptive adjustments. Therefore, in a broader sense, any game that allows users to gain benefits beyond entertainment can be

considered a serious game.

2.3.2 Application of Serious Games in Education

Education is a core application of serious games, providing learners with personalized, interactive, and engaging learning experiences through an "edutainment" approach. This method effectively addresses the issues of boredom or difficulty in traditional cybersecurity education, significantly enhancing learning motivation and participation. For example, the game "Permission Impossible" developed by Sehl et al. [48] helps players learn firewall strategies through drag-and-drop components, sparking interest in firewall concepts and deepening understanding. Similarly, Deeb et al. [23] researched a 3D escape room game that teaches cryptography and decryption knowledge. The mechanism of solving cryptographic challenges to escape from the room greatly stimulated players' initiative in learning and exploring.

2.3.3 Serious Game Development Framework

Le Compte et al. [34] proposed a development framework for serious games aimed at cybersecurity education, intended to help beginners effectively and efficiently learn and master cybersecurity knowledge. This framework, which aligns with the basic logic of the software lifecycle, is divided into six steps: Preliminary Analysis, Design, Development, Game Assessment, Deployment, and Player Assessment. The framework is based on the LM-GM model [35], and serious games developed using the framework have been proven to be able to maintaining good teaching outcomes by effectively mapping learning mechanics to game mechanics, even in informal contexts without marketing campaigns, advertisements, or demonstrations. However, the framework is more suitable for projects with longer development cycles and better staffing. Considering the limited time and resources for this project, the framework was combined with the waterfall development methodology [8], which is easy to manage and delivers results quickly. It was simplified and optimized into four phases: Analysis, Design, Development, and Evaluation, and was applied to this project accordingly.

2.3.4 Serious Game Design Model

The Learning Mechanics-Game Mechanics (LM-GM) model proposed by Theodore et al. [35] is a design and analysis model aimed at ensuring that learning mechanics are

properly integrated with game mechanics in serious games. As shown in Figure A.1 in Appendix A.1, the integration of Pedagogical Patterns and Game Design Patterns generates Serious Game Mechanics (SGM). SGM ensures that serious games maintain entertainment within game while also providing the functionality to teach knowledge and skills. As shown in Figure A.2 in Appendix A.1, this framework identifies 30 learning mechanics (on the left) and 36 game mechanics (on the right) through the reflection and summarization of both educational and game theories. Serious game developers can create an LM-GM map based on these two dimensions of mechanics and use the map to identify SGMs, highlighting the game and learning activities involved in the serious game mechanics. This approach enables SGMs to be transferred or further optimized across different serious games.

For example, the "Hypothesis" learning mechanic, which requires players to verify their hypotheses through experimentation, reflection, and analysis, can be combined with the "Strategy/Planning" game mechanic, which allows players to decide their action paths through strategic thinking. This combination can result in an SGM design where players need to conduct data analysis to formulate the correct strategy to complete the game's tasks. This SGM can focus on either strengthening reasoning and analytical skills, or mastering strategic planning skills in specific domains. However, as seen in the above case, while this model effectively finds a balance between educational functionality and entertainment, it does not reveal the relationship between specific SGMs and the higher-level instructional objectives that the serious game is intended to achieve.

2.4 Summary

This chapter has discussed issues related to web application security, the exploitation and defence of file upload vulnerabilities, and the current state of CTF-based cybersecurity education. While CTF competitions provide opportunities for practical learning, they are often challenging for beginners and lack sustained motivation. Subsequently, by exploring the concept, application, development framework, and design models of serious games, it was found that serious games offer advantages in terms of flexibility in adjusting the difficulty of knowledge and maintaining continuous user engagement. These findings provide insights into addressing the limitations of CTF-based teaching and offer theoretical support and methodological guidance for the design and development of serious games aimed at teaching file upload vulnerabilities.

Chapter 3

Requirements Gathering and Analysis

3.1 Objective-Driven Design Methodology

As discussed in Section 2.3.4, the LM-GM model has certain limitations when mapping SGM to high-level instructional objectives. To address this issue, the study proposes a methodology based on the LM-GM model. This approach is guided by predefined instructional objectives, collects requirements related to learning and game motives, and then identifies the corresponding LM and GM from the LM-GM model based on the collected requirements. The result is the creation of SGM that effectively serve the instructional objectives (Figure 3.1).

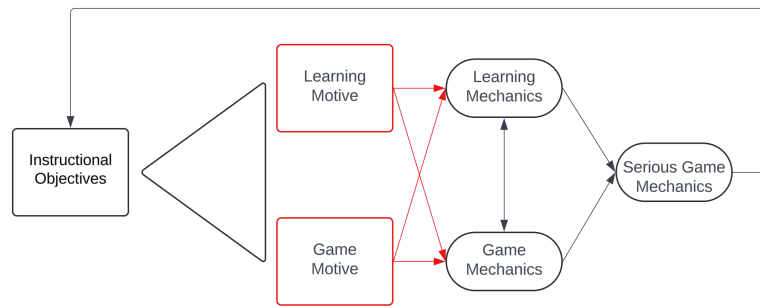


Figure 3.1: Objective-Driven Serious Game Design Framework Based on LM-GM Model

Instructional objectives refer to the knowledge, skills, or abilities that serious game developers or educators expect students to acquire through the game. Learning motives refer to users' preferences regarding the learning content, teaching strategies, and the difficulty of knowledge in relation to the instructional objectives. Game motives refer to users' preferences regarding the gaming experience, game type, and art style when faced with instructional objectives. Based on the users' learning and game motive

requirements, the most suitable LMs and GMs are selected from the LM-GM model to form the SGM, which are mapped to the instructional objectives. This ensures that the serious game effectively integrates educational and entertainment elements while meeting users' needs in the process of achieving the instructional objectives.

3.2 Survey on Learning and Game Motives

According to the Objective-Driven Design Methodology, before collecting user requirements related to learning and game motives, it is necessary to first clarify the instructional objectives and target users of the serious game.

Based on the project's goals, the instructional objectives are determined as follows: 1) Identifying of vulnerabilities: understanding which web application functionalities may lead to file upload vulnerabilities. 2) Exploiting vulnerabilities: mastering how to use discovered file upload vulnerabilities to execute further attacks. 3) Remediating vulnerabilities: learning how to modify affected web applications to defend against attacks related to file upload vulnerabilities.

The target users are undergraduate students, which can be further divided into the following three categories: 1) Those with no knowledge of web application security but with a foundation in computer theory and programming experience. 2) Those with some knowledge of web application security, having studied at least one course related to cybersecurity, but lacking practical experience. 3) Those who are very familiar with web application security, are cybersecurity majors, or have substantial practical experience in cybersecurity.

Based on the identified instructional objectives and target users, a questionnaire was designed to collect requirements related to learning and game motives to map the SGM to the instructional objectives.

As shown in Appendix B, the questionnaire consists of eight questions, divided into three sections: basic information, learning motives, and game motives. 1) The basic information section contains two questions about the participants' undergraduate year and their level of knowledge in web application security. 2) The learning motives section includes three questions regarding the participants' preferences on learning content, teaching strategies, and learning difficulty. 3) The game motives section contains three questions about preferences related to the gaming experience, game type, and art style.

The survey was conducted in online questionnaire form to facilitate distribution and data collection. A total of 10 respondents who met the characteristics of the target users

were recruited, and all respondents completed the questionnaire.

3.2.1 Questionnaire Results Analysis

In the basic information section, more than half of the respondents had knowledge of web application security, and all were in the final two years of their undergraduate studies.

In the learning motives section, users' interest in file upload vulnerability education was primarily focused on defence measures (60%) and vulnerability exploitation tools (50%), with the least interest in compliance and security standards (0%). Regarding learning strategies, hands-on practice and demonstrations of automated tools were the most favoured (70%), while documentation and tutorials were also preferred by half of the respondents. None of the respondents showed interest in high-difficulty file upload vulnerability content; instead, most preferred the medium difficulty.

In the game motives section, users' expectations for serious game for teaching file upload vulnerability were primarily focused on interaction that provides convenient controls and a good user experience (70%) as well as engaging gameplay closely aligned with learning objectives (60%). Preferences for game types were varied, with role-playing games(RPGs) standing out as the most favoured (30%). Regarding art style, pixel art was liked by nearly all participants, and 70% of participants expressed a preference for 2D art styles.

Overall, the results indicate that users prefer to learn about file upload vulnerabilities defence measures, and the use of automated exploitation tools through a role-playing game with a 2D pixel art style. With further analysis, target users hope to integrate knowledge with practice through a systematic learning path within the game, gaining advanced skills through the practical application of tools.

3.3 Case Study on LMs and Game GMs

To better design the SGMs, this project conducted in-depth case studies of LMs and GMs used in current web application security education. First, two mainstream web application security education platforms were analysed to understand their core LMs. Then, two serious games focused on cybersecurity education were examined to explore their LMs and GMs. These case studies aim to help the project identify effective mechanics and avoid ineffective ones.

3.3.1 Case Study of Web Security Education Platforms

A. Hack The Box Academy

Hack The Box Academy is a paid web security learning platform provided by Hack The Box (HTB) [30], offering systematic learning paths tailored for learners with different backgrounds and goals. These learning paths consist of multiple modules, allowing users to track their progress in real-time and review their performance on each module to better plan their learning. Each module includes several CTF Jeopardy-style challenges. Users must complete the current challenge before advancing to the next, progressing through the entire module.

During each challenge, HTB Academy provides basic vulnerability exploitation knowledge and hints, but this information is typically insufficient for completing the tasks directly. If users are unable to solve a challenge after a long period, they can opt to subscribe to an annual membership at a high fee to access the solutions or seek help on the HTB forum. This approach may discourage beginners who are unwilling to pay or who have invested significant time without success, potentially leading them to abandon the platform. Although the HTB forum provides a space for users to engage with and learn from one another, the accuracy of the information shared on the forum is not guaranteed, which could mislead users and hinder their understanding of the concepts. Furthermore, HTB modules focus primarily on vulnerability identification and exploitation, with limited opportunities for practicing vulnerability remediation.

B. PortSwigger Web Security Academy

PortSwigger Web Security Academy [49] is a free online platform provided by PortSwigger, the developer of Burp Suite, focusing on teaching web application security skills. The platform categorizes security topics into three levels: Server-side, Client-side, and Advanced. Each topic contains several CTF Jeopardy-style labs where users practice web application security by exploiting vulnerabilities to capture flags.

Unlike HTB Academy, PortSwigger's topics are designed with increasing difficulty, ranging from basic to complex. The labs are divided into three levels: APPRENTICE, PRACTITIONER, and EXPERT, helping users progressively learn through a step-by-step approach. There is no dependency between labs, allowing users to complete them in any order and adjust their learning paths according to their needs. Another significant difference is that PortSwigger provides detailed guidance for each lab, including step-by-step written solutions and video tutorials from the official community. This approach lowers the entry barrier for beginners learning through CTF-style challenges, avoiding

the frustration seen in HTB Academy where users struggle to complete challenges. However, the easy access to solutions may lead users to rely on them, reducing independent thinking and exploration, which could affect the overall learning experience. Similar to HTB Academy, PortSwigger also features a dashboard to track progress and a leaderboard to enhance user engagement and motivation.

Based on the analysis and comparison of the two web security teaching platforms, the following effective and ineffective learning mechanics can be identified:

	Effective	Ineffective
Learning Mechanics	• Community forum support	• Insufficient basic knowledge guidance
	• Modular learning paths	• Lack of opportunities for vulnerability defence practice
	• Progressive difficulty in knowledge settings	• Solutions are too easily accessible
	• Real-time feedback and support	• High complexity in hands-on practice steps
	• Flexible learning path options	
	• Detailed guidance and support information	
	• Learning progress tracking functionality	
	• Ranking system	

Figure 3.2: Web Security Education Platforms Case Study Result Table

3.3.2 Case Study of Cybersecurity Serious Games

A. CTFPICO 2013 - Toaster Wars

Toaster Wars [58] is a computer security game developed by the Carnegie Mellon University’s hacker team PPP. The game was created for the picoCTF computer security competition aimed at high school students. During the ten-day competition, participants had to solve 57 security challenges across various categories, including forensics, cryptography, and web security, distributed across four levels of the game.

As a narrative-driven single-player role-playing game, Toaster Wars features a multi-path storyline where players help a robot named Toast, who has crash-landed in a backyard, recover his spaceship and win the space hacking competition. Players unlock challenges by interacting with NPCs and using the Problem Viewer. Additionally, the game offers security knowledge lectures to assist participants without prior computer science or security experience in getting started. Ultimately, Toaster Wars successfully provided participants with the opportunity to learn and practice cybersecurity skills while experiencing the excitement and realism of CTF competitions. The game received positive feedback from both teachers and students.

	Learning Mechanics	Game Mechanics
Effective	<ul style="list-style-type: none">• Preliminary foundational knowledge teaching: This mechanic helps beginners without computer security experience build confidence and lays a foundation for more complex challenges later, effectively reducing the frustration that may arise during the learning process.• Milestone: Participants receive certificates after completing each stage, providing positive feedback at intervals, enhancing their sense of achievement, and motivating them to continue progressing through the learning material.	<ul style="list-style-type: none">• Multi-path story progression: After completing certain challenges, participants unlock the next stage of the story and new challenges. This prevents unfamiliar challenges from blocking progress and ensures smooth gameplay.• Leaderboard mechanic: This mechanic ranks all participating teams based on game points from highest to lowest, and the inclusion of physical prizes further boosts participant motivation.
Ineffective	<ul style="list-style-type: none">• CTF Jeopardy challenge: This mechanic focuses on attack techniques, leaving participants with little opportunity to practice defence techniques. Additionally, there is minimal introduction to foundational knowledge related to these techniques.	<ul style="list-style-type: none">• Continuous participation: This mechanic can lead to participant fatigue or resistance. The need for prolonged gameplay in CTF competitions may significantly reduce participant engagement and motivation in the later stages.

Figure 3.3: Toaster Wars Case Study Result Table

B. A NERD DOGMA

A NERD DOGMA [17] is a CTF Jeopardy-based escape room adventure game primarily designed for players with a foundational knowledge of computers but not specializing in cybersecurity. Unlike Toaster Wars, the game splits players into two roles: agents responsible for physical infiltration tasks and hackers who remotely decrypt and perform attacks through virtual terminals. Both roles must collaborate to obtain the password that disables the malicious software “A NERD DOGMA” released by the NERD Corporation.

To achieve this, players must solve cybersecurity challenges to retrieve the PIN codes for each room’s lock, and after unlocking three rooms, they can access the final room to obtain the password. The developers emphasize that cybersecurity games should allow players to use external systems or resources when necessary, rather than relying solely on in-game simulations. Upon its release, the game received a large amount of positive feedback and was praised for its appeal.

	Learning Mechanics	Game Mechanics
Effective	<ul style="list-style-type: none">• External tool integration: This mechanic allows participants to learn how to operate decryption tools during the game, helping them gain practical experience in combining knowledge with tool usage.	<ul style="list-style-type: none">• Task-driven narrative: This mechanic enhances immersion by engaging participants through role-playing and storyline progression, making players more invested in completing challenges and improving both engagement and the overall game experience.
Ineffective	<ul style="list-style-type: none">• Self-directed learning: This mechanic requires participants to independently explore and learn the knowledge needed to complete challenges. The lack of foundational guidance may discourage participants who have weaker learning abilities or lack initiative.	<ul style="list-style-type: none">• Challenge countdown: This mechanic requires participants to learn and apply relevant knowledge within a limited time. The time constraints may create excessive pressure, negatively impacting participants' ability to fully understand the content.• Multiplayer cooperation: While this mechanic enhances interaction among participants, it imposes significant limitations for solo players, restricting gameplay options.

Figure 3.4: A NERD DOGMA Case Study Result Table

3.4 Functional Requirements and Use Cases

Based on the analysis of user learning and game motives in the context of a serious game focused on teaching file upload vulnerabilities, combined with the case studies of mainstream cybersecurity education platforms and two typical serious games, the functional requirements for the serious game were summarized and proposed (Appendix A.2). From these functional requirements, use case diagrams for the game world system and Quest challenge system in the serious game were developed (Appendix A.3).

3.5 Summary

This chapter following the Objective-Driven Design Methodology Based on LM-GM, clarified the instructional objectives and identified the target users. Through user surveys and case studies, the chapter gathered the learning and game motives, as well as the LMs and GMs applicable to web application security education. Finally, these were further refined into the system functional requirements and use cases for the serious game, providing requirement guidance for the subsequent design and development of the serious game.

Chapter 4

Design

The design phase is guided by the requirements analysis and is based on the game design process elements concept proposed by Haltsonen et al. [31] . This section systematically unfolds the design of the serious game, covering game mechanics, gameplay flow, background story, and level design.

4.1 Serious Game Mechanics

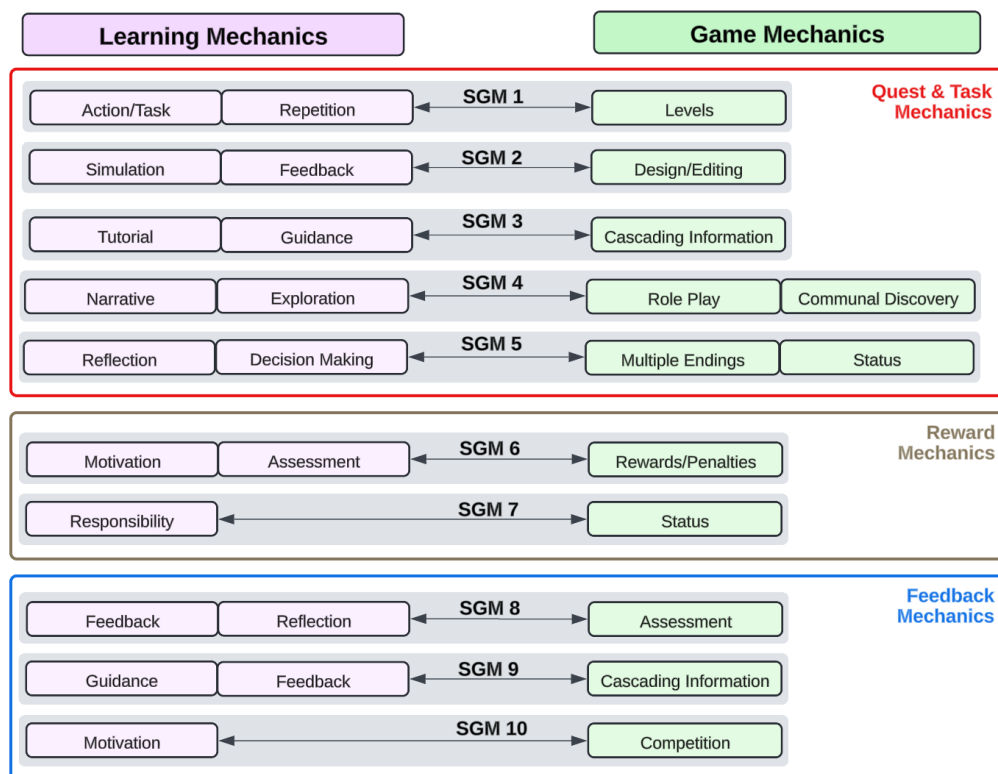


Figure 4.1: Mapping of LMs with GMs of SGMs

Based on the requirements gathered during the requirements analysis phase and drawing from case studies of learning and game mechanics, the project selected appropriate LMs and GMs from the LM-GM model's mechanics list (Figure A.2 in Appendix A.1), and eventually combined to form the 10 core serious game mechanics (SGM) applicable for the serious game, Neon City Defender. These SGMs are categorized into three types: task mechanics, feedback mechanics, and reward mechanics. The following sections will detail each SGM, along with the learning and game mechanics from the LM-GM model they utilize. Figure 4.1 illustrates the LM-GM map used for the serious game.

4.1.1 Task Mechanics

The task mechanics are designed in accordance with Csikszentmihalyi's Flow Theory [18], which emphasizes that the tasks users perform should match their abilities. This prevents tasks from being too difficult, leading to frustration and abandonment, or too easy, reducing the game's challenge, thereby maintaining players' sustained interest and engagement.

1) Progressive Difficulty in Challenges (LM: Action/Task, Repetition; GM: Levels)

The difficulty of the level challenges in the serious game will gradually increase. As players progress through the levels, they will steadily expand their knowledge of file upload vulnerability exploitation and defence. For example, players will first need to complete a relatively simple challenge involving a file upload vulnerability with no filtering or validation. Only after this will they unlock more difficult challenges, such as those involving blacklist filtering or additional defence mechanisms. This design ensures that players gain enough confidence in the early stages while continuing to advance in the game. Additionally, players can retry challenges multiple times to further deepen their understanding of the concepts and explore new solutions or strategies.

2) Vulnerability Remediation Practice (LM: Simulation, Feedback; GM: Design/Editing)

After completing vulnerability exploitation task in a quest challenge, players can engage in vulnerability remediation task. By modifying the code of the vulnerable application, players can attempt to fix the vulnerabilities. The system allows unlimited attempts, and after each submission of remediation code, the system provides feedback on whether the fix was successful. This feedback guides players in exploring the best remediation methods further. To prevent less experienced players from losing confidence due to getting stuck, the game allows progression to the challenge of next level even if the remediation task is incomplete. This mechanic helps players understand

vulnerability defence knowledge in a simulated environment and gain experience in fixing vulnerabilities.

3) Tutorial for Beginners (LM: Tutorial, Guidance; GM: Cascading Information)

When new players enter the game world system, a tutorial will guide them through the basic controls and the rules for using interface elements, ensuring they can smoothly enter the quest challenges. Additionally, when players enter the quest challenge system, the tutorial will explain the functions and usage of challenge page elements in detail, allowing players to quickly get started and focus on completing vulnerability exploitation and remediation tasks without spending excessive time learning interface components. The tutorial provides foundational and easy-to-understand information in a straightforward format, helping beginners avoid confusion due to unfamiliarity with the game environment.

4) Task-Driven Narrative (LM: Narrative, Exploration; GM: Role Play, Communal

Discovery) Each level quest in the serious game includes a small story, which not only stands independently but also connects to form a larger overarching plot. Players advance the main storyline by completing these tasks, motivating them to continue exploring the game to reveal the full story until the final ending. By role-playing, players become immersed in advancing the story, creating a learning experience that effectively maintains their interest.

5) Multiple-Endings Linear Storyline (LM: Reflection, Decision Making; GM:

Multiple Endings, Status) The game's storyline progresses in a linear fashion, with players experiencing quest events in a fixed sequence. However, the outcomes of vulnerability remediation tasks completed by players will impact the character's reputation, which directly influences whether the story has a good or bad ending. Players can only guide the game toward a positive ending by completing more vulnerability remediation tasks to improve their character's reputation. After reaching an ending, players automatically enter a second playthrough, allowing them to retry challenges to achieve different outcomes. This mechanic enhances replayability by introducing multiple endings while retaining a linear narrative, giving players opportunities to attempt advanced strategies as they accumulate knowledge and skills.

4.1.2 Reward Mechanics

The reward mechanics are designed based on Maslow's Hierarchy of Needs Theory [36]. According to Maslow, lower-level needs (such as physiological and safety needs)

lose their motivational effect once satisfied, quickly giving way to higher-level needs (such as esteem and self-actualization needs). Thus, when a serious game focuses on satisfying players' higher-level needs through rewards, their experience of value increases, which in turn boosts player engagement and a sense of achievement.

6) Player Scoring System (LM: Motivation, Assessment; GM: Rewards/Penalties)

The serious game will award different points based on players' performance in challenge tasks, such as completion time and the use of hints or solutions. Players who complete tasks faster and without using hints or solutions will receive higher scores. Additionally, the game features a points leaderboard that encourages players to complete levels as quickly as possible and accumulate more points to achieve higher rankings. By earning top scores and ranking high on the leaderboard, players not only experience the satisfaction of self-actualization but also gain respect and recognition from other users.

7) Character Reputation System (LM: Responsibility; GM: Status)

Players improve their in-game character's reputation by completing vulnerability remediation tasks. As more remediation tasks are completed, the character's reputation grows, leading to positive effects in the game world. By helping or saving NPCs in the game, players have the opportunity to achieve a perfect story ending. This mechanic encourages players to actively participate in more remediation tasks to learn more about vulnerability defence while realizing higher self-worth within the game world.

4.1.3 Feedback Mechanics

The feedback mechanics are designed based on the Fogg Behavior Model [27], which identifies motivation, ability, and prompt as essential for behavior to occur. When motivation or ability is low, the model suggests using incentives to boost motivation or reducing obstacles to improve ability. With both in place, the right prompt triggers the behavior. To better provide prompts to players, the feedback mechanics are designed to operate across three dimensions: historical, immediate, and global.

8) Historical Feedback: Learning Progress Tracking (LM: Feedback, Reflection;

GM: Assessment) Players can review their performance in past quest challenges by viewing challenge history. This includes scores, completion status of vulnerability exploitation/remediation tasks, completion time, number of hints used, and whether solutions were viewed. This information helps players understand their learning progress and performance. By reviewing these records, players can assess their current knowledge and skill levels, which motivates them to retry challenges to achieve better results.

9) Immediate Feedback: Subtask Completion Recognition (LM: Guidance, Feedback; GM: Cascading Information) During vulnerability exploitation tasks, the objectives section will provide sequential subtasks required to complete the vulnerability exploitation. As players complete each subtask, the system immediately marks it as done. This immediate feedback allows players to track their progress and understand how far they are from completing the overall task. If players struggle to complete a step after multiple attempts, they can view hint information to get assistance and successfully complete the exploitation. By breaking down complex tasks into smaller subtasks, this mechanic provides accomplishment for each subtask completed, encouraging players to continue advancing, thereby gradually improving their skills and building confidence.

10) Global Feedback: Points Leaderboard (LM: Motivation; GM: Competition) The serious game features a global leaderboard based on player scores, displaying the top twenty players as well as the current player's total score and ranking. This mechanic fosters a competitive atmosphere among players, motivating them to perform better in challenges to achieve higher scores and rankings. When players see others surpassing them on the leaderboard, they feel a sense of competition, which drives them to continuously challenge themselves, enhance their skills, and learn more advanced knowledge and techniques.

4.2 Gameplay Flow

The gameplay flow for this serious game was designed based on the SGMs. As shown in Figure 4.2, after registering and logging in, new players receive an introduction to the background story and operation tutorial. Once in the game world, players explore and switch scenes to enter different levels. By interacting with NPCs, they accept quests and start challenges. Before challenges, players can examine hint items to gather the knowledge needed for quest challenge.

In challenges, players exploit file upload vulnerabilities to capture the flag, then fix the vulnerability by modifying the application's code. Completing the exploitation unlocks the next level, while completing the fix boosts their reputation. Players must complete the exploitation task of the challenge to unlock the next level. After finishing all levels, the player's reputation determines a good or bad ending. If players exit a challenge, they return to the game world, where they can view challenge history, leaderboards, character reputation, or exit to the login page.

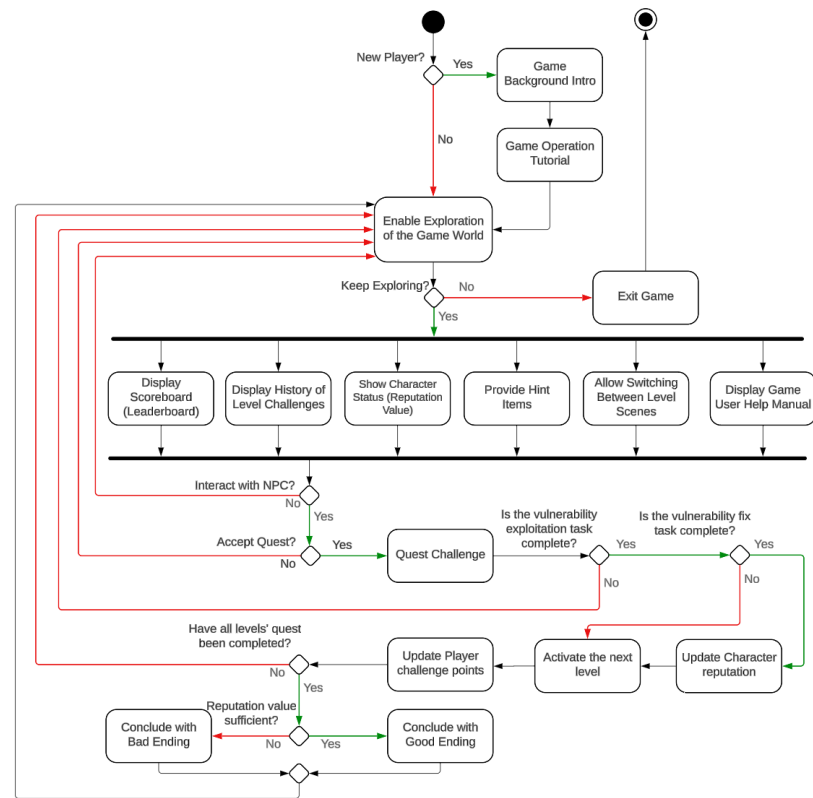


Figure 4.2: Activity Diagram for Neon City Defender Gameplay Flow

4.3 Game Background Story

The game is set in 2177 in Neon City, a megacity controlled by powerful tech corporations. Players take on the role of Alex Mercer, a cybersecurity enthusiast using hacking to expose corporate misconduct and assist oppressed citizens. Alex helps Nova Castell search for her missing mother, Lyra, starting by exploiting a vulnerability in Lyra’s personal website that reveals her involvement in a secret Militech experiment. Alex then aids Drake Vex, a former Arasaka employee, by exploiting vulnerabilities in Arasaka’s cyberware trading site to restore his cyberware legs. Drake reveals that Lyra was taken by NCPD agent Jack Morgan. Alex hacks the NCPD’s cyberware tracking database, locating Lyra through her tracking chip. The game’s ending, determined by the player’s reputation, dictates whether Alex rescues Lyra.

Considering user preferences for 2D pixel RPGs and limited development time, the game’s theme was set as Cyberpunk. The popularity of Cyberpunk 2077 [15] released in 2020, showcased a dystopian society of “high tech, low life,” addressing themes like social inequality and corporate control—resonating with real-world concerns.

Additionally, the widespread availability of open-source Cyberpunk-themed tile sets [56][57] and character assets [9][43] significantly reduced development time.

4.4 Game Levels

4.4.1 Level Challenge Page Design

The game features three levels using the same quest challenge page framework. As shown in Figure 4.3, when players click "Access Challenge," the quest page opens in their default browser. Players must first complete the exploitation task (red box) to unlock the remediation task (green box). In both sections, players submit their answers: the flag for exploitation and the modified code for remediation.

The screenshot shows a web browser window with the URL <http://websecurity.com>. The page is titled "Quest Challenge" and has buttons for "Help" and "Leave Challenge".

File Upload Level-1

Scenario Description
Content of the challenge background description

Task
Content of the task description 1000 Pts ⓘ

Access vulnerability APP Access IP: 10.10.124.35 Time Remain: 59 min 59 sec Rest

Enter the FLAG here Submit

Objectives

- ☐ Sub-Task 1
- ☒ Sub-Task 2
- ☐ Sub-Task 3
- ☐ Sub-Task 4
- ☐ Sub-Task 5

Guidance

Cheat Sheet

Hint

Solution

Defence Section

Defencen Scenario Description
Content of the Defencen challenge background description

Task
Content of the task description 500 Pts ⓘ

index.html Check.php Shop.php

```
1 <!doctype html>
2 <html>
3 <head>
4   <!-- Internal game scripts/styles, mostly boring stuff -->
5   <script src="/static/game-frame.js"></script>
6   <link rel="stylesheet" href="/static/game-frame-styles.css" />
7
8   <!-- Load jQuery -->
9   <script
10     src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js">
11   </script>
12 </head>
```

Update

Objectives

Hint

Solution

```
<?php
header('Content-Type: application/json');

// Check if a file has been uploaded
if (isset($_FILES['avatar'])) {
    // Get the uploaded file information
    $file = $_FILES['avatar'];
    $uploadDir = 'uploads/'; // Set the upload directory
    $allowedTypes = ['image/jpeg', 'image/png', 'image/gif', 'application/msword',
```

Figure 4.3: User Interface Design for Level Challenge Page

During the exploitation task, clicking “Access Vulnerability Application” generates a Docker container running the vulnerable application, which opens in a new tab. The container’s address and remaining time are displayed, and players can reset it if needed. In the remediation task, players can view and edit code, then click “Update” to replace backend files. As the applications use interpreted PHP, changes take effect immediately without recompilation. The system automatically tests the fix, and players can retry the exploitation task to verify the remediation, enhancing their understanding of the vulnerability.

4.4.2 Level Design

Each level features a quest challenge where players must first exploit, then fix, a file upload vulnerability. The levels progressively increase in difficulty, starting with a basic file upload vulnerability and culminating in a complex challenge involving both file upload and local file inclusion vulnerabilities. To address the demand for advanced exploitation skill practice identified in Section 3.2.1, the level design incorporates mainstream web security tools. Players will use Wappalyzer [55] to identify the web application technology stack and Burp Suite [45] to intercept and modify requests, perform fuzz testing, and exploit file upload vulnerabilities.

A. Level 1 (Level Name: Lab Rat)

Level 1 introduces Unrestricted File Upload vulnerabilities. During exploitation, players locate the file upload feature, identify the technology stack using Wappalyzer, write and upload a web shell, and use it to execute linux commands that retrieve the password in the password.txt file. During remediation, players modify the upload function’s PHP code to restrict the upload file extensions through a whitelist filter and can implement double-extension checking or other validations for more advanced protection.

B. Level 2 (Level Name: Shattered Limbs)

Level 2 introduces validation flaws in file uploads and server misconfigurations. Building on Level 1, this level incorporates blacklist validation for file extensions, whitelist validation for Content-Type, and introduces the ability to overwrite the .htaccess file to rewrite file parsing rules.

During exploitation, players need to identify the file upload feature, confirm the server’s technology stack, and use Burp Suite to perform fuzz testing to discover allowed file extensions and Content-Types. After finding that .htaccess files can be uploaded,

players upload one to configure specific file extensions to be parsed as PHP code, write a web shell, and use it to execute commands to retrieve the activation code stored on the web server. In remediation, players modify the PHP code to blacklist .htaccess files or can implement whitelist validation for file extensions to enhance security.

C. Level 3 (Level Name: Omnipresent Surveillance)

Level 3 focuses on file upload and local file inclusion vulnerabilities. Building on Level 2, this level introduces whitelist validation for file extensions, Content-Type, and file signatures, and addresses the Apache server configuration flaws in Level 2. It also adds local file inclusion vulnerabilities due to insufficient request parameter validation.

In exploitation, players can use Burp Suite for fuzz testing to identify allowed file extensions, Content-Type, and file signatures. When direct exploitation fails, they need to discover the local file inclusion vulnerability by modifying request path parameters. Players then can upload a web shell, execute it to retrieve database connection info, and write a malicious script to extract data from the database. In remediation, players need to implement whitelist validation for request parameters to block non-page files and add file name blacklist filtering to prevent uploads with names matching website pages, enhancing protection against file tampering or replacement.

4.5 UI Design

To ensure the Neon City Defender's user interface has high usability, Nielsen's Ten Heuristics [42] were applied to optimize the interface design, focusing primarily on four key areas: Visibility of System Status, Match Between the System and the Real World, User Control and Freedom, and Flexibility and Efficiency of Use. Detailed implementation of these optimizations can be found in Appendix A.4.

4.6 Summary

This chapter guided by the requirements, detailed the design process of ten SGMs categorized into tasks, rewards, and feedback based on the LM-GM model. These SGMs were integrated into the gameplay flow and a Cyberpunk-themed background story, introducing three progressively difficult levels within the story. Finally, Nielsen's Heuristic Principles were used to optimize the UI design, ensuring high usability. The overall design provides a comprehensive framework and implementation plan for the development of the Neon City Defender.

Chapter 5

Implementation

This chapter describes the system architecture, user interface, key technologies, and the implementation of core SGMs in the serious game.

5.1 System Architecture of the Serious Game

To enhance development efficiency and prevent changes in one part of the code from affecting multiple areas of the project, the serious game employs a frontend-backend separation architecture. This modular approach reduces coupling and allows the frontend and backend subprojects to be developed and modified independently. Additionally, to ensure compatibility across various programming languages and frameworks, all system requests are based on platform-independent RESTful HTTP protocols.

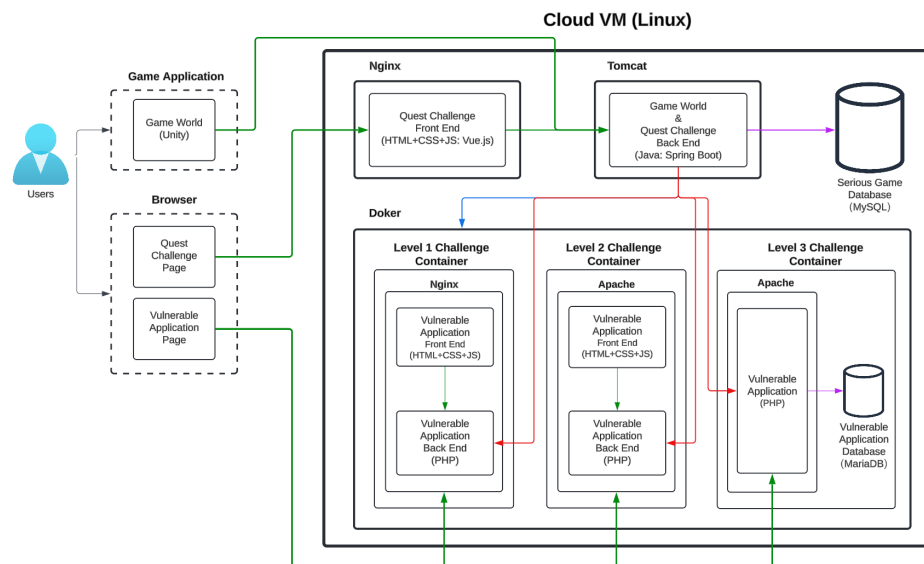


Figure 5.1: System Architecture of the Serious Game

As shown in Figure 5.1, the backend uses the LTMJ technology stack (Linux + Tomcat + MySQL + Java), while the frontend utilizes Vue.js [53] to implement the quest challenge pages, and the game client application is developed using the Unity [51] engine. Compared to the more common LAMP architecture (Linux + Apache + MySQL + PHP) [5], the use of Java in the LTMJ stack enables the introduction of the Spring Boot framework [50], which is well-suited for building decoupled and modular architectures, which facilitates frequent modifications to different functional components during development. Additionally, Tomcat's multi-threading capabilities and support for high concurrency ensure that the game can handle multiple concurrent player requests smoothly, optimizing the gaming experience even with limited computational resources.

Figure 5.1 illustrates four types of system communication channels: client-to-server requests (green arrow), database communication (purple arrow), Docker API communication (blue arrow), and internal VM network communication (red arrow). These channels respectively handle data interaction between the game client/browser and the backend, data management between the backend and the database, container management between the backend and Docker, and internal network communication between the backend and the vulnerable applications within the containers, ensuring seamless cooperation and data synchronization among system components. Appendix A.5 details the implementation and functionality of each communication by types.

5.2 Cloud Deployment

As shown in Figure 5.1, aside from the game client application, which must run locally on the player's system, the Vue frontend for quest challenges, the Spring Boot backend, the MySQL database, and the three vulnerable application images used in the challenges are all deployed on an Alibaba Cloud [3] Linux virtual machine. This deployment eliminates the need for players to configure a complex local environment and install tools (e.g., Docker, MySQL, IP address configuration, port settings). Players can simply run the .exe file included in the game installation package on their Windows system to launch and play the game directly. This deployment approach offers players an optimal gaming experience, simplifies the initialization process, and effectively avoids technical issues related to local configuration. It ensures that players can run the game smoothly and focus on the content and learning about file upload vulnerabilities. Furthermore, the pre-configuration of the cloud server, database, vulnerable application images, and system environment guarantees stability and consistency for all players in the game

environment, providing a solid foundation for subsequent evaluation.

5.3 Serious Game User Interface

Initially, the game client was planned to run as a WebGL application in the browser. However, since the challenges and vulnerable applications need to open in new pages, this could make it difficult and confused for users to navigate and operate across multiple browser tabs. To improve the user experience, the game was packaged as a standalone application client, ensuring that users can switch between pages more intuitively and smoothly during gameplay. The user interface implementation based on the gameplay flow is shown in Figure 5.2. More serious game UI can be found in Appendix G.

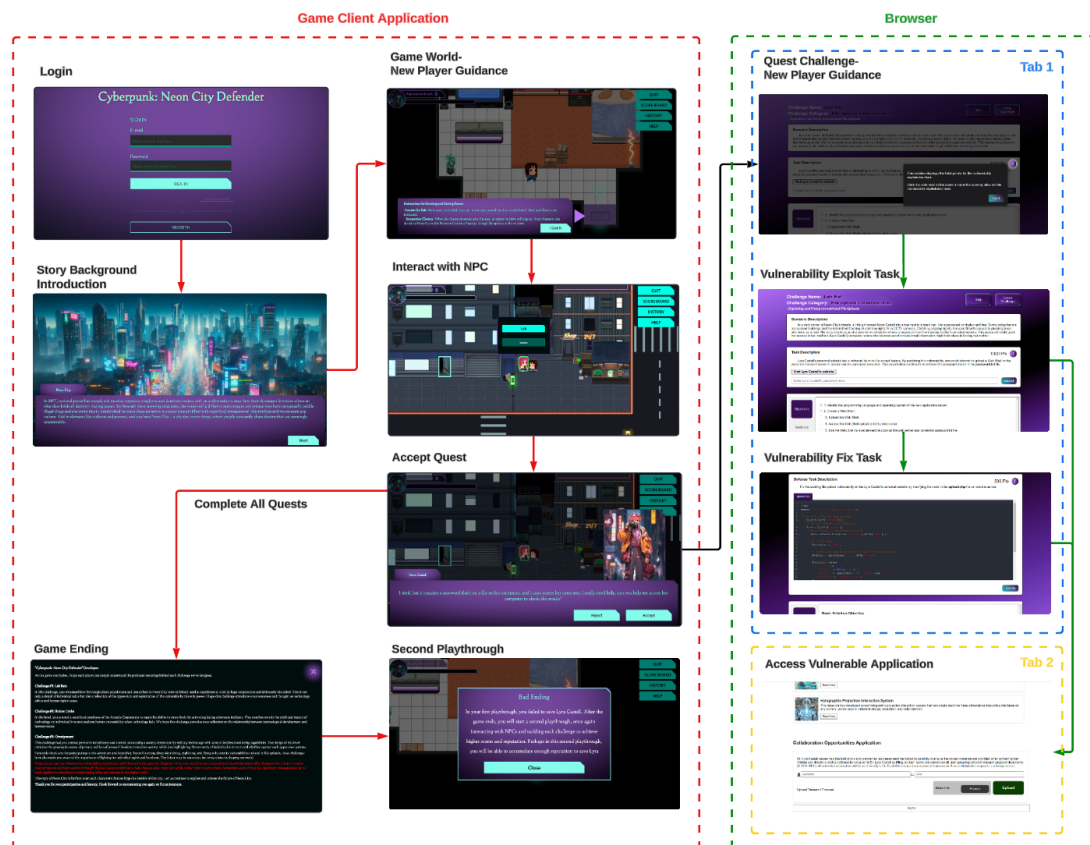


Figure 5.2: User Interface Implementation for Neon City Defender

5.4 Core Technologies Applied

Vue.js [53] is a JavaScript frontend framework that uses the MVVM (Model-View-ViewModel) pattern. By separating the View (user interface), Model (data), and View-

Model (the connection between View and Model), MVVM decouples data, business logic, and the user interface. The two-way data binding in MVVM automatically synchronizes data between the Model and View, enabling real-time updates of code changes made in the vulnerability remediation editor (View) to the Model. The ViewModel sends the modified code (Model) to the backend and updates the Model based on the returned remediation results, allowing the user to see remediation feedback immediately. The two-way binding in MVVM ensures real-time user operations, making it ideal for scenarios that require instant feedback, such as vulnerability remediation tasks.

Unity [51] is a cross-platform game development engine that offers user-friendly visual editing features. Developers can easily build scenes and logic by dragging and dropping components and modifying properties, significantly shortening the learning curve. Although Unity's rendering capabilities may not be as powerful as the Unreal Engine, it also demands less from the hardware. Given that target undergraduate students playing the serious game may need to run it on various hardware setups, and considering the game's 2D pixel art style, Unity provides sufficient rendering capabilities with lower resource consumption, making it a more suitable development tool for this project.

Spring Boot [50] is a Java-based web application development framework that simplifies application setup with features such as automatic configuration, an embedded Tomcat server, and streamlined dependency management. With Maven handling dependencies, developers can easily integrate third-party libraries, such as the Java JWT library for JSON Web Tokens and the Docker Java Core library for Docker interaction, speeding up the development process. Additionally, Spring Boot projects can be packaged as standalone JAR files, enabling easy deployment on any system with a Java environment, which facilitates later cloud deployment.

MySQL [40] is an open-source, high-performance, and stable relational database that seamlessly integrates with the Spring Boot framework through Spring Data. With the interface-based programming model provided by Spring Data JPA, developers can easily perform data operations on MySQL by simply defining the repository interfaces provided by Spring Data, without needing to write complex SQL queries.

Nginx [41] is a cross-platform, lightweight web server developed in C that includes reverse proxy capabilities. It effectively isolates sensitive backend teaching resources and user data in the serious game. Users only need to access the public-facing Nginx server, and all requests are forwarded by Nginx to the backend web server for processing, concealing the real IP address of the backend server. This architecture reduces the security risks of directly exposing the backend server and enhances the internal network

security of the serious game’s cloud virtual host.

Docker [25] is a containerization technology that offers faster startup times and lower resource consumption compared to traditional virtual machines. By sharing the host kernel, Docker can achieve near-native performance. For the serious game involving multiple vulnerable applications, Docker provides container-level process isolation, ensuring that containers running the vulnerable applications are independent from the cloud virtual host. Even if a web application vulnerability is exploited, the attacker can only affect the application within the current container, leaving the host system and other containers unharmed. This enhances overall system security. Additionally, this aligns with the principle of least privilege(PoLP) in information security, ensuring that applications or processes only receive the minimum privileges required for their operation, effectively reducing the attack surface.

5.5 Implementation of Core SGMs

5.5.1 Subtask Completion Recognition

The implementation of the subtask completion recognition SGM consists of two parts: frontend polling to update subtask status and backend logic for subtask recognition.

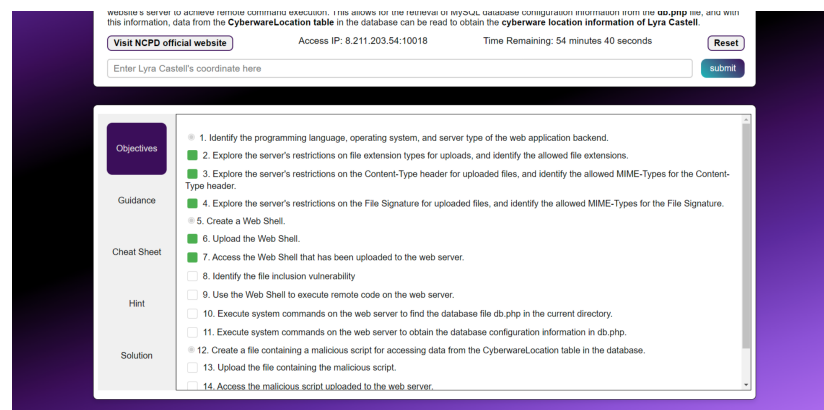


Figure 5.3: Subtask Completion Recognition for Vulnerability Exploitation Tasks

On the frontend, the subtask recognition functionality is integrated into the “Objectives” menu component within the exploitation section of the challenge page (Figure 5.3). Each level contains a list of different subtasks, and completed subtask status indicator will turn green. Before all subtasks are completed, the frontend sequentially checks the status of each subtask. The checkObjective() method shown in Figure A.6

On the backend, each level has its own subtask recognition logic. To facilitate expansion and maintenance, the subtask recognition process is abstracted into a class called `CheckpointStrategy`. The specific recognition logic for each level is implemented by subclasses of this abstract class. By separating the common recognition process from the specific subtask recognition logic, all levels can reuse the same recognition process framework, which improves code reusability and development efficiency while reducing functional coupling. This provides a solid foundation for expanding future levels. As shown in Figure A.7 in Appendix A.6, the `executeCheck()` method in the `CheckpointStrategy` class first determines whether the subtask needs recognition. If it does, it calls the abstract `executeTask()` method implemented by the subclass to execute the specific recognition logic. For example, in the first level, the `executeTask()` method queries the Nginx logs to check for requests that contain `cmd=find`, `cmd=ls`, or `cmd=ll`, along with `password.txt`, to determine whether the player has completed the subtask of finding the `password.txt` file.

The implementation of the vulnerability remediation practice SGM can be divided into four stages: frontend code retrieval, backend code replacement, vulnerability remediation validation, and vulnerability application reset.



As shown in Figure 5.4, during the frontend code retrieval stage, the page integrates the CodeMirror editor module, allowing users to view, modify, and submit vulnerable code. When the editor is initialized, the code from backend file is mounted to the DOM for user editing. After the user clicks the “UPDATE” button, the modified code and corresponding file name are packaged and sent to the backend. Once the backend completes the remediation validation and responds, the result is displayed to the user.

In the backend code replacement stage, the backend receives the modified code from the frontend, generates the target URL based on the player’s current container IP and port, and sends the code to the container via a POST request. As shown in Figure A.8 in Appendix A.6, after the container receives the code, it retrieves the file information and generates the file path based on the file name, replacing the original backend file using the `move-uploaded-file()` method.

In the vulnerability remediation validation stage, the system performs simulated attack test cases to detect the remediation status. Each level has its own independent test cases, and the remediation is deemed successful if all test cases pass. If any test case fails, the remediation fails. For example, in the second level, the system checks for the remediation of a server misconfiguration vulnerability by detecting the upload of an `.htaccess` file. As shown in Figure A.9 in Appendix A.6, the system first calls the `deleteAccFiles()` method to delete `.htaccess` files in the container, then re-uploads the file and checks the container’s file list. If the `.htaccess` file is not present, the remediation is successful. Additionally, after all test cases are completed, the system deletes any temporary files generated by the simulated attacks.

During the vulnerability application reset stage, after the vulnerability remediation validation is completed, the system resends the original files preset in the backend to the container, replacing the files modified by the user. This restores the vulnerable application to its initial state, ensuring consistency for the next round of remediation.

5.6 Summary

This chapter outlined the development of the serious game system using a frontend-backend separation architecture and an LTMJ backend framework, introduced the concepts behind the user interface implementation, and detailed the implementation of two core SGMs: subtask completion recognition and vulnerability remediation practice. The cloud deployment ensures accessibility and stability, laying a solid foundation for the subsequent evaluation phase.

Chapter 6

Evaluation

This chapter evaluates the functionality, usability, and teaching effectiveness of the serious game.

6.1 Functionality Evaluation

A.Objectives and Methods

The functionality evaluation aimed to verify if the core features of the serious game met user requirements and provided a stable experience. Functional testing was conducted based on system requirements derived from user needs and case study analysis in Section 3.3. Each functional test case (Appendix C) outlined objectives, input conditions, and expected outcomes to assess success. The testing environment mirrored the development environment, and each test was executed manually with results recorded.

B. Results and Analysis

All functional test cases were validated, confirming that core features like scene switching, leaderboard viewing, accessing vulnerable applications, and task submission performed as intended. This demonstrated stable system functionality, laying a solid foundation for subsequent user evaluations.

6.2 Evaluations Involving User Research

The user research evaluations focused on usability and teaching effectiveness. Usability was assessed through semi-structured interviews [32] and a SUS questionnaire [7],

while teaching effectiveness was evaluated with pre- and post-tests. All participants completed the evaluations, with details provided in Sections 6.3 and 6.4.

6.2.1 Participants

Due to time constraints, only three participants matching the three target user categories identified in Section 3.2 were found for the study. Although Woolrych's research [29] suggests that five participants are typically required to uncover most usability issues, the diverse backgrounds of the three participants, representing the three target user categories, enabled a relatively comprehensive analysis by comparing their similarities and differences.

For reference, participants are coded as P1, P2, and P3: **P1**: A third-year student with basic cybersecurity knowledge, has studied software development, databases, and computer science theory. **P2**: A fourth-year student with minimal cybersecurity knowledge, familiar with algorithms and machine learning, but with limited programming skills. **P3**: A fourth-year computer science student with deep knowledge of cybersecurity, and practical experience in web application security and penetration testing.

6.2.2 User Research Process

The usability and teaching effectiveness evaluations were conducted simultaneously. After participants were briefed on their rights, data usage, and signed consent forms (Appendix I), the research began. Participants first completed a 10-minute pre-test, followed by 1 hour of gameplay. After, they took a 10-minute post-test, participated in a 30-minute semi-structured usability interview, and completed the SUS questionnaire. The entire process took around 2 hours.

6.3 Usability Evaluation

The usability evaluation aims to assess users' experience while playing the serious game, ensuring that the game reduces the entry barrier to build players' confidence while also continuously motivating and engaging them. According to ISO 9241-11 [10], usability is characterized by how effectively, efficiently, and satisfactorily particular users can accomplish their objectives within a certain context when interacting with a product. To analyse and draw conclusions effectively, the usability evaluation will be

conducted using semi-structured interviews and the SUS questionnaire, focusing on the three dimensions of effectiveness, efficiency, and satisfaction. By combining the depth of semi-structured interviews with the breadth of the SUS questionnaire, the evaluation seeks to comprehensively measure the usability of the serious game.

6.3.1 Semi-Structured Interviews

A.Objectives and Methods

The semi-structured interviews [32] aim to gather in-depth user feedback and suggestions regarding the usability of the serious game. This feedback will be used to analyse and identify which SGMs in the game exhibit good usability, effectively helping players build confidence and motivation, as well as identify areas for improvement based on participants suggestions.

Before the semi-structured interviews, participants played the serious game naturally without any assigned tasks to ensure their behaviour was not influenced. To effectively collect user feedback on usability and gather improvement suggestions, the interview was structured around nine core questions (see Appendix D), including three closed questions to gather basic user information and six open-ended questions focusing on overall impressions of the serious game, specific feature usage experiences, and learning experiences. After the interviews, the recordings were transcribed, and thematic qualitative analysis methods [11] were applied to the textual data.

B.Results and Analysis

During the analysis, relevant textual data were identified and categorized into seven codes. These codes were grouped into three themes: ‘Teaching Content’, ‘Learning Methods’, and ‘Game Experience’. As shown in Figure 6.1, The themes were then mapped to the three usability dimensions: effectiveness, efficiency, and satisfaction.

Usability Categories	Codes	Themes
Satisfaction	Knowledge Gained	Teaching Content
Effectiveness	Guided Instructions	Learning Methods
	Learning Through Practice	
	Subtask Guidance	
Efficiency	Game Difficulty	Game Experience
	Sense of Achievement	
Satisfaction	Game Style	

Figure 6.1: Mapping of Usability Categories to Codes and Themes

Theme 1: Teaching Content

The teaching content theme is summarized from the ‘Knowledge Gained’ code, which mainly reflects participants’ satisfaction with the teaching content of the serious game. All three participants provided positive feedback on the knowledge they gained. P1 mentioned that during the vulnerability exploitation task in level 2, he reviewed the file signature knowledge that he had not fully understood during a previous cybersecurity course. P2 stated that the hints in the defence section of the game made him realize the importance of mastering programming languages in code remediation. P3 highlighted that he learned to use Burp Suite through the fuzz testing for vulnerability identification, something he had been eager to master. He emphasized, “After mastering Burp Suite in level 2, I was eager to use it in level 3.”

Theme 2: Learning Methods

The learning methods theme is summarized from the codes ‘Guided Instructions’, ‘Learning Through Practice’, and ‘Subtask Guidance’. They mainly reflect participants’ feedback on whether the learning methods were effective.

In terms of ‘Guided Instructions’, P2 and P3 emphasized that the tutorial provided clear guidance that helped them quickly integrate into the game. However, P2 noted that the game operation tutorial was overly verbose. In terms of ‘Learning Through Practice’, P1 and P3 reported gaining new knowledge through hands-on code modification during vulnerability remediation tasks, specifically deepening their understanding of content-type rules and expanding their PHP syntax knowledge. However, P2 struggled to learn from code modification, noting that submitting ineffective remediation code did not help confirm the correctness of the knowledge he was practicing, and his lack of programming experience made it hard to understand the remediation principles in the solutions. In terms of ‘Subtask Guidance’, all participants praised the subtask completion recognition feature in the vulnerability exploitation section. P2 mentioned that the subtask lists helped him effectively identify unfamiliar knowledge from the guidance.

Theme 3: Game Experience

The game experience theme includes ‘Game Difficulty’, ‘Sense of Achievement’, and ‘Game Style’. While “Game Difficulty” and “Sense of Achievement” relate to efficiency, “Game Style” primarily reflects satisfaction.

In terms of ‘Game Difficulty’, P2 and P3 found the level 1 challenge reasonable but noted a sharp difficulty increase between levels 1 and 2. P2 struggled with Burp Suite in level 2, while P1 suggested some hints were not closed enough to the subtasks, making exploitation more difficult. In terms of ‘Sense of Achievement’, P2 stated that the joy

of completing subtasks encouraged him to continue challenging more subtasks. P3 said combining leaderboard with challenge history helped him identify which challenges he did not perform well, which pushed him to revisit challenges for more points. In terms of ‘Game Style’, participants gave positive feedback. P1 found the storyline engaging, particularly with multiple endings, prompting a second playthrough. P2 liked the cyberpunk and pixel art style, while P3 enjoyed the music but suggested adding an option to turn it off during challenges.

Thematic Qualitative Analysis Result

The thematic qualitative analysis of Neon City Defender gathered feedback on eight of the ten core SGMs, highlighting their role in building player confidence and motivation, as well as areas for improvement.

For satisfaction, users satisfied the knowledge gained, the storyline, and the art design. The introduction of automated vulnerability exploitation tools enhanced confidence for future challenges. Task-driven narratives and multi-ending storylines boosting engagement. Regarding effectiveness, the tutorial and subtask completion recognition SGMs effectively lowered the learning curve, enabling quick gameplay integration. However, the vulnerability remediation practice SGM requires more detailed explanations to support users with weaker foundations. In terms of efficiency, the points system, progress tracking, vulnerability remediation, and subtask recognition SGMs motivated continued participation in challenges. While the progressive difficulty provided a reasonable starting point, additional levels are needed to ensure smoother transitions.

6.3.2 SUS Questionnaire

A.Objectives and Methods

The System Usability Scale (SUS) [7] is a quick tool for evaluating product usability, consisting of 10 Likert-scale questions and suitable for participants with diverse backgrounds. In this project, SUS supplemented the usability evaluation from the semi-structured interviews. After the interviews, participants completed the SUS questionnaire (Figure E.1 in Appendix E). To assess usability more intuitively, the SUS questions were grouped by three categories: Effectiveness (Questions 4, 5, 7, 10), Efficiency (Questions 2, 3, 7, 8), and Satisfaction (Questions 1, 3, 6, 9).

B.Results and Analysis

Using the SUS scoring method, total scores for each participant were calculated, along with effectiveness, efficiency, and satisfaction scores. These were then converted

to a 0-100 scale, as shown in Table 6.1.

Participants	Score			
	SUS Overall	Effectiveness	Efficiency	Satisfaction
P1	80.0%	75.0%	75.0%	87.5%
P2	72.5%	68.8%	68.8%	87.5%
P3	82.5%	75.0%	81.3%	93.8%
Avg.	78.33%	72.92%	75.00%	89.58%

Table 6.1: Participant SUS Scores and Usability Breakdown

Based on the SUS evaluation standards from Figure A.10 in Appendix A.7, the average SUS score of 78.33% exceeded the typical 68%, meeting the ‘GOOD’ rating standard, indicating overall satisfactory usability.

Effectiveness scores ranged from 68.8% to 75.0%, reflecting stable performance in aiding users to learn and operate, consistent with the qualitative analysis highlighting the beginner tutorial and subtask recognition SGMs as confidence boosters. Efficiency scores showed more variance, with P2 scoring only at the average level, suggesting that aspects like difficulty progression and the extensive tool operation guidance may have caused slowdowns in the gameplay flow. Satisfaction scores were notably high, with all participants scoring above the ‘EXCELLENT’ rating, aligning with the positive feedback on learning content and game design. Notably, beginner P2 scored lower in Efficiency and Effectiveness compared to others, reinforcing the qualitative analysis that pointed out the negative impact of insufficient explanations in vulnerability remediation solutions and steep difficulty progression on beginners.

Overall, the SUS questionnaire analysis closely aligns with the qualitative results from Section 6.3.1, with high satisfaction ratings but room for improvement in Effectiveness and Efficiency

6.4 Teaching Effectiveness Evaluation

A.Objectives and Methods

To evaluate the teaching effectiveness of the serious game in identifying, exploiting, and remediating file upload vulnerabilities, pre- and post-tests [13] were used. Pre-tests assessed participants’ baseline knowledge before playing, while post-tests evaluated their knowledge gains afterward. Comparing these results demonstrated participants’

progress and the game's effectiveness in achieving instructional objectives.

To further assess changes in knowledge and skills related to vulnerability identification, exploitation, and remediation, nine standardized test questions were designed based on instructional objectives in Section 3.2. The tests, administered in written form, used the same set of questions (Appendix F).

B.Results and Analysis

Participants	Pre-test Accuracy	Post-test Accuracy	Difference
P1	44.4%	77.8%	33.3%
P2	22.2%	44.4%	22.2%
P3	66.7%	100.0%	33.3%

Table 6.2: Pre-test and Post-test Accuracy

As shown in Table 6.2, all three participants improved their test accuracy after playing the serious game. A t-test [38] was conducted to assess whether this improvement is statistically significant, rather than due to random chance. Although the small sample size of three participants may introduce a margin of error, the results can still serve as preliminary reference points for future research that can validate these findings by expanding the sample size.

The t-test calculated the difference, sample size, and standard deviation, resulting in a t-value of 7.97 and a corresponding p-value of 0.015. This indicates that, with a sample size of three, the improvement in post-test scores is statistically significant ($p < 0.05$), suggesting that the serious game positively impacted participants' mastery of file upload vulnerability knowledge and skills, showing good teaching effectiveness.

6.5 Summary

This chapter confirmed the stable operation of the serious game through functional testing, laying the groundwork for user research. Usability evaluation, combining semi-structured interviews and the SUS questionnaire, revealed high user satisfaction with content and design, demonstrating that the game effectively lowered entry barriers to build confidence, and maintained motivation. However, there remains room for improvement in efficiency and effectiveness. The teaching effectiveness evaluation showed significant gains in users' knowledge of file upload vulnerabilities, confirming positive teaching effectiveness, though a larger sample size is needed for further validation.

Chapter 7

Conclusion

7.1 Overview

This project developed Neon City Defender, a serious game aimed at teaching undergraduate students how to identify, exploit, and remediate file upload vulnerabilities in web application security. By incorporating CTF Jeopardy-style gameplay with vulnerability remediation practice SGM, enable students to practice both exploitation and remediation theoretical knowledge in real-world scenarios.

The project adhered to the LM-GM serious game design model, using Learning LMs and GMs to create balanced SGMs that combined functionality with engagement. To address the limitation of insufficient alignment between SGMs and instructional objectives, the requirements for students' learning and game motives were gathered for identifying LMs and GMs during the requirements phase using the proposed objective-driven design methodology based on the LM-GM model. This ensured that the SGMs effectively served achievement of the instructional objectives. Additionally, insights from current cybersecurity platforms and serious game case studies refined system requirements and use cases.

During the design phase, the project developed ten core SGMs categorized into tasks, rewards, and feedback based on the identified LMs and GMs. These SGMs were used to structure a complete gameplay flow, with three progressively challenging levels designed around a Cyberpunk theme. In the implementation, a decoupled front-end and back-end system architecture was built, integrating the LTMJ framework (Linux + Tomcat + MySQL + Java), ensuring the SGMs were effectively implemented while maintaining stability and accessibility via cloud deployment. The evaluation phase involved functional testing, semi-structured interviews, SUS questionnaires, and

pre- and post-tests, which validated the serious game's effectiveness in teaching the identification, exploitation, and remediation for file upload vulnerabilities.

Overall, Neon City Defender performed as expected, demonstrating strong usability, lowering the entry barrier to build student confidence, and sustaining motivation. While the overall usability received positive feedback with high ratings in satisfaction, there is still room for improvement in efficiency and effectiveness. Moreover, although Neon City Defender helped students master the knowledge and skills of file upload vulnerabilities through gameplay, further research with larger sample sizes is needed to fully validate its teaching effectiveness.

7.2 Further Work

Based on the qualitative analysis from Section 6.3.1, the main usability issue in terms of efficiency was the heavy reading load, leading to delays during challenges and lowering overall gameplay efficiency. Reducing the difficulty gap between levels and incorporating video media could alleviate this issue by providing information more intuitively. In terms of effectiveness, beginners with weaker foundations struggled to grasp remediation logic, even after reviewing solutions, affecting their engagement and completion rates. Adding detailed code comments and foundational knowledge explanations could help them better understand remediation and apply it in later challenges.

Security is a key concern, as all system communication currently uses unencrypted HTTP, posing a risk of exposing sensitive data like users' email addresses and login credentials to potential MITM attacks. To mitigate this, implementing HTTPS is essential to safeguard user data against unauthorized access or theft.

Regarding game functionality, adding a forum feature could enhance the game's social aspects by providing a platform for users to exchange insights on vulnerability exploitation and defense. This would encourage collaboration and fulfill higher-level needs in Maslow's hierarchy [36]. Sharing experiences and achievements would offer users a sense of accomplishment, motivating further progress in the game.

In terms of evaluation methods, the small sample size of three participants in the t-test for teaching effectiveness could lead to significant error, reducing the stability and reliability of the conclusions. Increasing the participant count would enhance representativeness and statistical significance, providing a more accurate assessment of Neon City Defender's teaching effectiveness.

Bibliography

- [1] Threats and vulnerabilities in web applications 2020–2021. <https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020-2021/>, 2022. [Accessed 1-08-2024].
- [2] Clark C Abt. *Serious Games*. University press of America, 1987.
- [3] Alibaba. Alibaba cloud introduction. <https://edu.alibabacloud.com/course/387>. [Accessed 20-06-2024].
- [4] Daniele Antonioli, Hamid Reza Ghaeini, Sridhar Adepu, Martin Ochoa, and Nils Ole Tippenhauer. Gamifying ics security training and research: Design, implementation, and results of s3. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, pages 93–102, 2017.
- [5] Amazon AWS. What is a lamp stack? <https://aws.amazon.com/what-is/lamp-stack/?nc1=hls>, 2024. [Accessed 30 – 06 – 2024].
- [6] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
- [7] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.
- [8] Youssef Bassil. A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*, 2012.
- [9] BattleInkMaps. 100 cyberpunk characters for dnd and game art. <https://battleinkmaps.itch.io/100-cyberpunk-characters-for-dnd-and-game-art>, 2024. [Accessed 15-07-2024].

- [10] Nigel Bevan, James Carter, and Susan Harker. Iso 9241-11 revised: What have we learnt about usability since 1998? In *Human-Computer Interaction: Design and Evaluation: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part I 17*, pages 143–151. Springer, 2015.
- [11] Virginia Braun and Victoria Clarke. *Thematic analysis*. American Psychological Association, 2012.
- [12] J Brooke. Sus: A “quick and dirty” usability scale. *Usability Evaluation in INdustry/Taylor and Francis*, 1996.
- [13] Jacalyn E Bryan and Elana Karshmer. Assessment in the one-shot session: Using pre-and post-tests to measure innovative instructional strategies among first-year students. *College & Research Libraries*, 74(6):574–586, 2013.
- [14] Miriana Calvano, Federica Caruso, Antonio Curci, Antonio Piccinno, Veronica Rossano, et al. A rapid review on serious games for cybersecurity education: Are” serious” and gaming aspects well balanced? In *IS-EUD Workshops*, 2023.
- [15] CDPR. Cyberpunk 2077. https://cyberpunk.fandom.com/wiki/Cyberpunk_2077, 2024. [Accessed 15-07-2024].
- [16] Kevin Chung and Julian Cohen. Learning obstacles in the capture the flag model. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.
- [17] Gabriele Costa, Martina Lualdi, Marina Ribaudó, and Andrea Valenza. A nerd dogma: Introducing ctf to non-expert audience. In *Proceedings of the 21st Annual Conference on Information Technology Education*, pages 413–418, 2020.
- [18] Mihaly Csikszentmihalyi and Judith LeFevre. Optimal experience in work and leisure. *Journal of personality and social psychology*, 56(5):815, 1989.
- [19] ctftime team. What is capture the flag? <https://ctftime.org/ctf-wtf/>, 2015. [Accessed 05-07-2024].
- [20] NATIONAL VULNERABILITY DATABASE. Cve-2024-40425 detail. <https://nvd.nist.gov/vuln/detail/CVE-2024-40425>, 2024. [Accessed 01-07-2024].

- [21] NATIONAL VULNERABILITY DATABASE. Cve-2024-40549 detail. <https://nvd.nist.gov/vuln/detail/CVE-2024-40549>, 2024. [Accessed 01-07-2024].
- [22] NATIONAL VULNERABILITY DATABASE. Cve-2024-40550 detail. <https://nvd.nist.gov/vuln/detail/CVE-2024-40550>, 2024. [Accessed 01-07-2024].
- [23] Fatima Abu Deeb and Timothy J Hickey. Teaching introductory cryptography using a 3d escape-the-room game. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–6. IEEE, 2019.
- [24] Damien Djaouti, Julian Alvarez, and Jean-Pierre Jessel. Classifying serious games: the g/p/s model. In *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*, pages 118–136. IGI global, 2011.
- [25] docker. Use containers to build, share and run your applications. <https://www.docker.com/resources/what-container/>: :text=A [Accessed 05-07-2024].
- [26] edgescan. 2024 vulnerability statistics report. <https://www.edgescan.com/intel-hub/stats-report/>, 2024. [Accessed 21-06-2024].
- [27] Brian J Fogg. Fogg behavior model. *Behav. Des. Lab., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 2019.
- [28] Vitaly Ford, Ambareen Siraj, Ada Haynes, and Eric Brown. Capture the flag unplugged: an offline cyber competition. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 225–230, 2017.
- [29] Kevin Godby. Why you only need to test with five users, 2012.
- [30] Ryan Gordon. Introduction to htb academy. <https://help.hackthebox.com/en/articles/5272936-introduction-to-htb-academy>, 2024. [Accessed 28-07-2024].
- [31] Jukka Haltsonen. Guide to writing a game design document. 2015.
- [32] Hanna Kallio, Anna-Maija Pietilä, Martin Johnson, and Mari Kangasniemi. Systematic methodological review: developing a framework for a qualitative semi-

- structured interview guide. *Journal of advanced nursing*, 72(12):2954–2965, 2016.
- [33] Rae Yule Kim. The impact of covid-19 on consumers: Preparing for digital sales. *IEEE Engineering Management Review*, 48(3):212–218, 2020.
- [34] Alexis Le Compte, David Elizondo, and Tim Watson. A renewed approach to serious games for cyber security. In *2015 7th International Conference on Cyber Conflict: Architectures in Cyberspace*, pages 203–216. IEEE, 2015.
- [35] Theodore Lim, Maira B Carvalho, Francesco Bellotti, Sylvester Arnab, Sara De Freitas, Sandy Louchart, Neil Suttie, Riccardo Berta, and Alessandro De Gloria. The lm-gm framework for serious games analysis. *Pittsburgh: University of Pittsburgh*, 2015.
- [36] Saul McLeod. Maslow’s hierarchy of needs. *Simply psychology*, 1(1-18), 2007.
- [37] Jelena Mirkovic and Peter AH Peterson. Class {Capture-the-Flag} exercises. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.
- [38] Prabhaker Mishra, Uttam Singh, Chandra M Pandey, Priyadarshni Mishra, and Gaurav Pandey. Application of student’s t-test, analysis of variance, and covariance. *Annals of cardiac anaesthesia*, 22(4):407–411, 2019.
- [39] Steve Morgan. Cybersecurity jobs report: 3.5 million unfilled positions in 2025. <https://cybersecurityventures.com/jobs/>, 2023. [Accessed 15-07-2024].
- [40] MySQL. Why mysql? <https://www.mysql.com/>, 2024. [Accessed 02-07-2024].
- [41] nginx. Basic http server features. <https://nginx.org/en/>, 2024. [Accessed 28-06-2024].
- [42] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 373–380, 1992.
- [43] NYKNCK. Cyberpunk characters. <https://nyknck.itch.io/cyberpunk-character>, 2019. [Accessed 15-07-2024].

- [44] Karishma Pooj and Sonali Patil. Understanding file upload security for web applications. *International Journal of Engineering Trends and Technology*, 42(7):342–347, 2016.
- [45] portswigger. Getting started with burp suite. <https://portswigger.net/burp/documentation/desktop/getting-started>, 2024. [Accessed 20-07-2024].
- [46] Hendri Pratama, Mohamed Nor Azhari Azman, Gulzhaina K Kassymova, and Shakizat S Duisenbayeva. The trend in using online meeting applications for learning during the period of pandemic covid-19: A literature review. *Journal of Innovation in Educational and Cultural Research*, 1(2):58–68, 2020.
- [47] Ngo Van Quyen. Hands-on training for mitigating web application vulnerabilities. 2023.
- [48] Sibylle Sehl and Kami Vaniea. Permission impossible: Teaching firewall configuration in a game environment. In *3rd European Workshop on Usable Security*, 2018.
- [49] Dafydd Stuttard. Introducing the web security academy. <https://portswigger.net/blog/introducing-the-web-security-academy>, 2019. [Accessed 28-07-2024].
- [50] VMware Tanzu. Spring boot. <https://spring.io/projects/spring-boot>, 2024. [Accessed 07-07-2024].
- [51] Unity. Getting started with unity. <https://unity.com/>, 2024. [Accessed 28-06-2024].
- [52] Giovanni Vigna, Kevin Borgolte, Jacopo Corbetta, Adam Doupé, Yanick Fratantonio, Luca Invernizzi, Dhilung Kirat, and Yan Shoshitaishvili. Ten years of {iCTF}: The good, the bad, and the ugly. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.
- [53] vue.js. What is vue? <https://vuejs.org/guide/introduction.html>, 2024. [Accessed 03-07-2024].
- [54] Jan Vykopal, Valdemar Švábenskỳ, and Ee-Chien Chang. Benefits and pitfalls of using capture the flag games in university courses. In *Proceedings of the 51st ACM Technical symposium on computer science education*, pages 752–758, 2020.

- [55] wappalyzer. Identify technologies on websites. <https://www.wappalyzer.com/>, 2024. [Accessed 19-07-2024].
- [56] WinLu. Winlu cyberpunk tileset - interior. <https://winlu.itch.io/winlus-cyberpunk-tileset-1-interior>, 2020. [Accessed 15-07-2024].
- [57] Winlu. Winlu cyberpunk tileset - exterior. <https://winlu.itch.io/cyberpunkexterior>, 2021. [Accessed 15-07-2024].
- [58] Kaiyang Zhang, Shihao Dong, Guoliang Zhu, Danielle Corporon, Tim McMullan, and Salvador Barrera. picoctf 2013-toaster wars: When interactive storytelling game meets the largest computer security competition. In *2013 IEEE International games innovation conference (IGIC)*, pages 293–299. IEEE, 2013.

Appendix A

Essential Supporting Materials

A.1 The LM-GM Serious Games Design Model

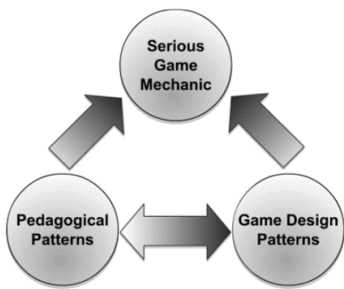


Figure A.1: LM-GM Framework [35]

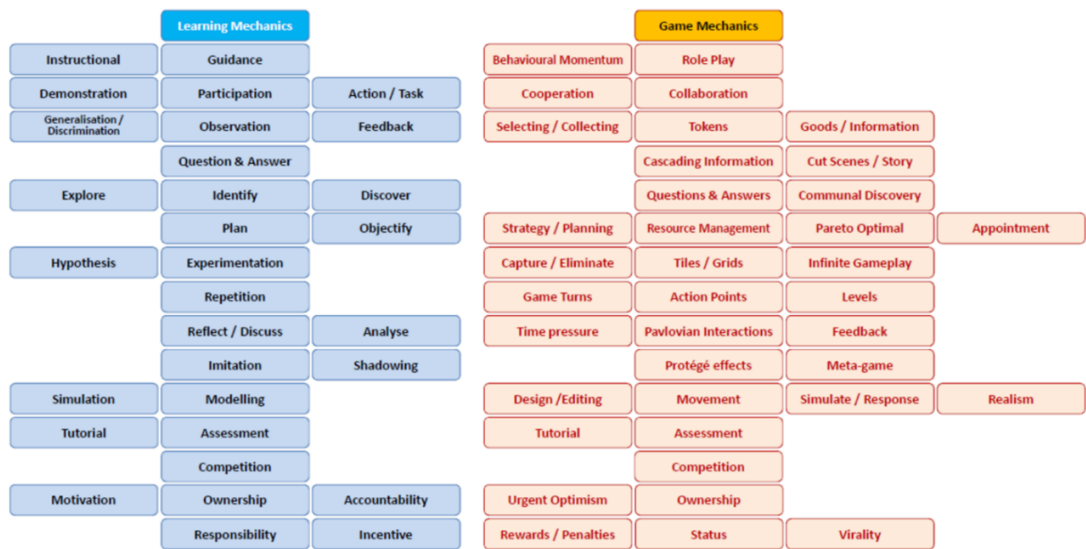


Figure A.2: Learning and Game Mechanics Lists of LM-GM Model [35]

A.2 Serious Game System Functional Requirements

System Functional Requirements Lists

Game World System

1. The system should support user registration for new accounts and login for existing accounts.
2. The system should allow users to control characters to move and explore the game world after a successful login.
3. The system should provide necessary background information before entering the game world.
4. The system should allow users to obtain quest information and learn the storyline through conversations with NPCs.
5. The system should allow users to accept quests through conversations with NPCs and start challenges of the quests.
6. The system should allow users to reject quests through conversations with NPCs and be able to accept them again later if desired.
7. The system should support users in obtaining helpful information for completing challenges by checking hint items.
8. The system should allow users to switch between different scenes in the game world.
9. The system should allow users to view the leaderboard to obtain score and ranking information.
10. The system should allow users to view character attributes to understand the growth of their player character.
11. The system should allow users to view challenge history to access history records of completed challenges.
12. The system should allow users to view the user help manual to get guidance on game operations.
13. The system should support users in exiting the game and resuming progress when they return.

Quest Challenge System

14. The system should support users in viewing challenge scenario descriptions to get the quest story background for the challenge.
15. The system should support users in checking descriptions of vulnerability exploitation and fix tasks, clarifying the objectives of the tasks.
16. The system should allow users to access vulnerable web applications to perform exploitation tasks.
17. The system should support users in resetting vulnerable web applications to retry exploitation tasks.
18. The system should support users in submitting answers for exploitation and fix tasks and provide feedback on success or failure.
19. The system should allow users to view guidance for exploitation and fix tasks, providing basic knowledge for completing the tasks.
20. The system should allow users to view sub-step information and corresponding hints for exploitation and fix tasks.
21. The system should allow users to view the solutions for exploitation and fix tasks.
22. The system should provide a corresponding fix task after completing the exploitation of a vulnerability.
23. The system should allow users to view and modify the code of the vulnerable application to complete fix tasks.
24. The system should support users in submitting answers for fix tasks and provide feedback on success or failure.
25. The system should allow users to view task scoring rules to understand the scoring criteria and requirements for challenges.
26. The system should allow users to exit the current challenge at any time.

A.3 UML Diagrams for Serious Game Use Cases

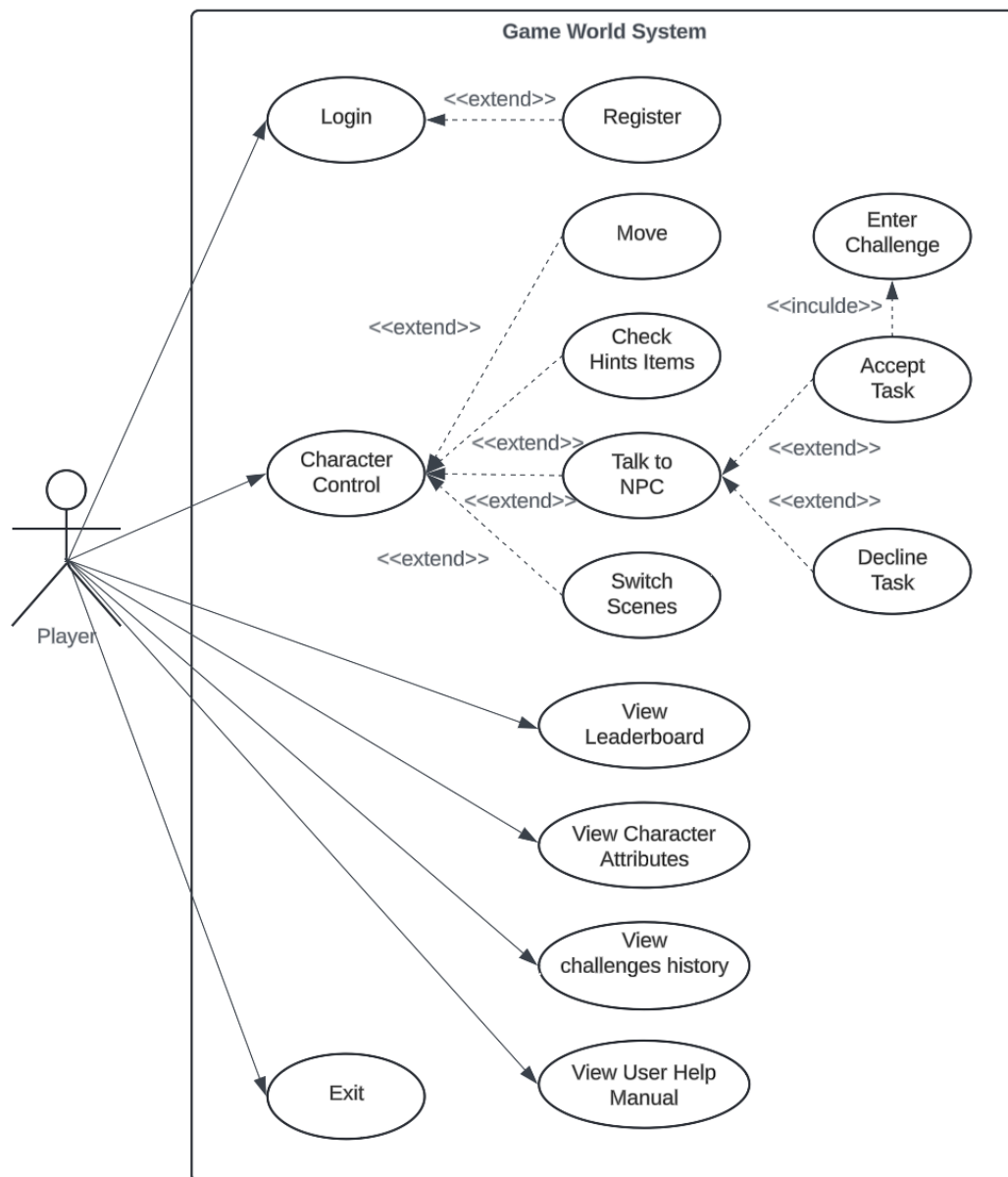


Figure A.3: Use Case Diagram for Game World System of SG



Figure A.4: Use Case Diagram for Quest Challenge System of SG

A.4 UI Optimizations using Nielsen's Ten Heuristics

1. Visibility of System Status

After players successfully complete the exploitation and remediation challenges, the system will display feedback showing the time taken and the score earned, allowing players to easily understand their performance. Additionally, during the remediation task, after players submit modified code, the system requires time to run simulated

attacks to determine if the fix was successful. During this time, a loading animation will be displayed, clearly indicating that the system is processing the request, thereby preventing confusion while waiting for feedback.

2. Match Between the System and the Real World

All pop-up windows in the system, when requiring binary choices from the user, consistently use red for negative options (e.g., No/Cancel/Back) and bluish-purple for positive options (e.g., Yes/Confirm/Stay). This design aligns with real-world user expectations, reducing the learning curve and improving the flow and ease of decision-making during interactions.

3. User Control and Freedom

When users attempt to exit a quest challenge or view the solution for the exploitation/remediation tasks, the system prompts a confirmation window. Exiting a challenge will result in progress loss and require restarting, while viewing the solution will prevent the player from earning points for the current task. This confirmation step effectively prevents irreversible consequences due to accidental actions, significantly boosting user confidence during gameplay.

4. Flexibility and Efficiency of Use

Upon entering the first level challenge, users can choose to skip the tutorial for the quest challenge system via a pop-up window, allowing returning players to bypass the tutorial. This provides a more flexible onboarding experience. Additionally, when a new level is unlocked, the system displays directional arrows in the game world pointing to the entrance of the next scene, greatly reducing the time users spend searching for the next level. This design allows players to focus more on the challenges and improves the system's overall efficiency.

A.5 Communication in the Serious Game System

A.5.1 Client-to-Server Requests (Green Arrow)

The quest challenge frontend project, built with Vue.js, is packaged as static files and deployed on an Nginx server hosted on a virtual machine. When players request the

challenge page through their browser, Nginx returns the corresponding file resources to the browser and presents them to the player. Additionally, Nginx serves as a reverse proxy server, receiving requests made by the challenge page via the Axios library and forwarding them to the serious game backend running on the embedded Tomcat server within Spring Boot. This setup facilitates data retrieval and updates for players, as well as synchronization of the challenge page content, such as submitting answers for exploitation tasks, submitting remediation code, and tracking subtask completion statuses. On the other hand, the game client uses the UnityWebRequest library to bypass Nginx's reverse proxy and directly communicate with the serious game backend to handle functions such as leaderboard management, challenge history records, and NPC status updates (Figure A.5) within the game world.

A.5.2 Database Communication (Purple Arrow)

To maintain and manage data used by the game client and Quest challenge frontend, the serious game backend communicates with the MySQL database using the Spring Data framework to handle data storage and retrieval. By leveraging Spring Data's interface-based CRUD operations (Create, Read, Update, Delete), the need to manually write SQL statements for database operation logic is eliminated, which improves development efficiency and code maintainability.

A.5.3 Docker API Communication (Blue Arrow)

The serious game backend communicates with the Docker daemon on the virtual machine via the Docker API to manage and monitor the vulnerable application containers. The backend first pings the Docker TCP interface exposed on the virtual machine to check connectivity. If the connection is successful, the backend server creates a Docker client object to send command requests and perform container creation, deletion, and query operations. This dynamic management of the container lifecycle ensures that each player's vulnerable application container can be flexibly created and destroyed as needed. For instance, containers are created when players enter a challenge and destroyed when they exit. Additionally, the vulnerable application images deployed on the cloud virtual machine correspond to the three challenge levels, with each image capable of generating containers with different internal environments to simulate different file upload vulnerabilities featured in the challenges.

A.5.4 Internal VM Network Communication (Red Arrow)

The game backend interacts with the PHP backend of the vulnerable applications running on the containers by using the RestTemplate class. The PHP backend, in addition to implementing the file upload functionality for the vulnerable applications, is responsible for log monitoring and viewing directory files within the application. Log monitoring is used to confirm whether players have successfully initiated specific requests, helping determine if they have completed subtasks in the exploitation challenge. Viewing directory file names is used to detect whether specific files have been updated or deleted, aiding in verifying whether the player has successfully implemented the vulnerability fix.

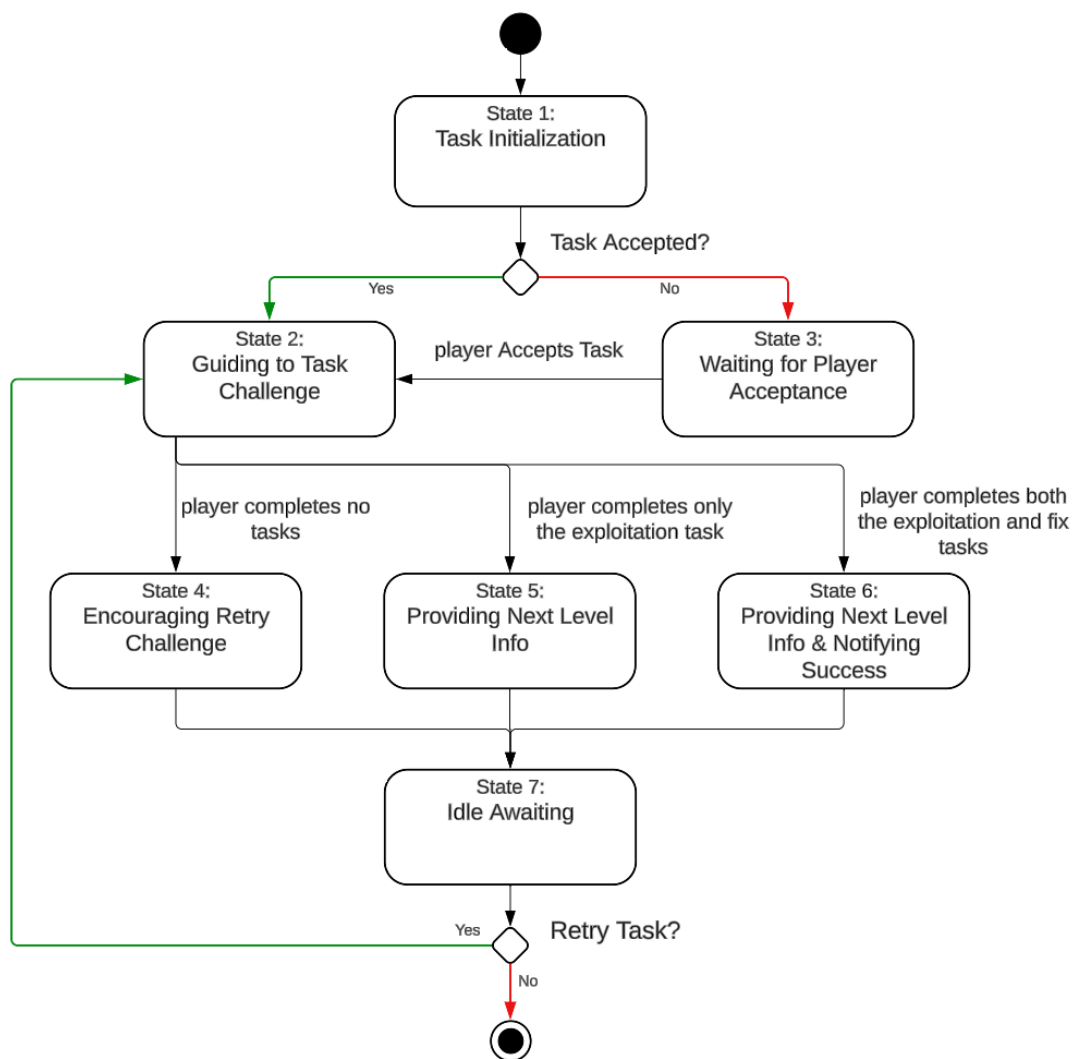


Figure A.5: State Diagram for State-Driven NPC's Quest Flow

A.6 Code Snippets of Core SGMs Implementation

```
const checkAllObjectives = async () => {
  for (let i = 0; i < selectedSteps.value.length; i++) {
    await checkObjective(i);
    await new Promise((resolve) => setTimeout(resolve, 1000));
  }
  if (!allStepsCompleted.value) {
    setTimeout(checkAllObjectives, 1000);
  }
}
```

Figure A.6: Asynchronous Polling Code for Front-end Subtask Completion Status Update

```
public Boolean executeCheck(Integer objectNumber, Boolean objectIsNeedCheck) {
  try {
    if (objectIsNeedCheck.equals(false)) {
      return true;
    }
    return executeTask(objectNumber);
  } catch (Exception e) {
    log.info(e.getMessage());
    return false;
  }
}

! usage 3 implementations Wenjia
abstract Boolean executeTask(Integer objectNumber);
```

Figure A.7: Abstract Method for Back-end Subtask Completion Status Check

```
} else if ($request == '/uploadDefence'
    && $_SERVER['REQUEST_METHOD'] == 'POST'
    && isset($_FILES['file'])) {
  $uploadDir = '/var/www/html/';
  $file = $_FILES['file'];
  if ($file['error'] != UPLOAD_ERR_OK) {
    echo json_encode([
      'code' => 400,
      'msg' => 'File upload error'
    ]);
    exit;
  }
  $fileName = basename($file['name']);
  $filePath = $uploadDir . $fileName;
  if (move_uploaded_file($file['tmp_name'], $filePath)) {
    echo json_encode([
      'code' => 200,
      'msg' => 'File uploaded successfully',
      'data' => [
        'filename' => $fileName,
        'filepath' => $filePath
      ]
    ]);
  }
}
```

Figure A.8: Code for Replacing Remediation Files in Vulnerable Application Containers

```
DockerDT0 dockerDT01 = dockerService.checkThenGetDockerIpPort();
remoteRestFulService2.deleteAccFiles(dockerDT01);
remoteRestFulService2.uploadFileByOneCheckPointApi(dockerDT0);

List<String> challengeCatalogueAllFileName = remoteRestFulService2.getChallengeCatalogueAllFileName(dockerDT0);
DefenceResult defenceResult = new DefenceResult();
if (!challengeCatalogueAllFileName.contains(".htaccess")) {
    defenceResult.setSuccess(Boolean.TRUE);
    defenceResult.setMessage("Vulnerability fix successful");
}
```

Figure A.9: Simulated Attack Testing Code for Vulnerability Remediation Verification

A.7 System Usability Scale Evaluation Standards

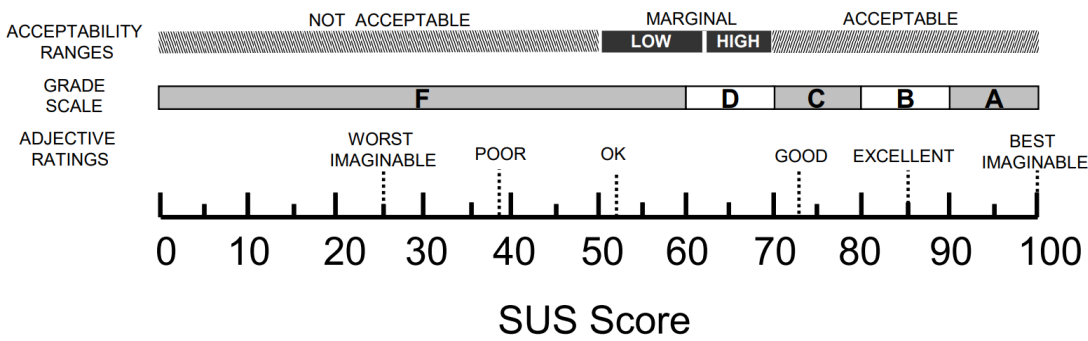


Figure A.10: SUS Score Comparison Chart with Adjective Ratings, Acceptability Ranges, and Average SUS Scores [6]

Appendix B

Questionnaire for Collecting Learning and Game Motives

Serious Game Requirements Gathering Questionnaire

Thank you for participating in this survey! Your feedback will help us develop a more effective serious game to teach knowledge related to web application security, especially about file upload vulnerabilities.

1. What year of undergraduate study are you currently in?
 - Year 1
 - Year 2
 - Year 3
 - Year 4
2. Which of the following categories best describes you?
 - I have no knowledge of web application security but have a background in computer theory and programming experience.
 - I have some knowledge of web application security, having studied at least one course related to cybersecurity, but I lack practical experience.
 - I am very familiar with web application security; I am either a cybersecurity major or have extensive practical experience in cybersecurity.
3. Which of the following topics related to file upload vulnerabilities in web application security are you interested in? (You may select multiple options)

- Types of file upload vulnerabilities
- Common attack methods
- Defense measures
- Exploitation tools
- Vulnerability detection and assessment methods
- Compliance and security standards
- Other: _____

4. Which of the following teaching methods do you think are suitable for teaching file upload vulnerability knowledge? (You may select multiple options)

- Theoretical explanation
- Hands-on practice
- Case studies
- Problem-solving
- Group discussions
- Demonstration of automated tools
- Online testing and assessment
- Learning through documentation and tutorials
- Other: _____

5. If a serious game were developed to teach file upload vulnerability knowledge, which type of game would you prefer it to be? (You may select multiple options)

- Puzzle games
- Simulation games
- Adventure games
- Strategy games
- Role-playing games
- Narrative games
- Other: _____

6. What difficulty level do you expect for a serious game that teaches file upload vulnerability knowledge?
- High difficulty: Requires memorizing a large amount of information and completing challenges through in-depth thinking, complex knowledge application, and technical practice.
 - Medium difficulty: Requires moderate information retention, reasonable thinking, and balanced knowledge application and technical practice to complete the challenges.
 - Low difficulty: Requires minimal memorization and simple thinking, allowing for basic knowledge application and technical practice to progress easily.
7. If you were to learn file upload vulnerability knowledge through a serious game, which aspects of the game would you value most? (You may select multiple options)
- Game interface: Clear, visually appealing, and easy-to-understand interface design
 - Interaction: Convenient controls and a good user experience
 - Game aesthetics: Visual effects, sound design, animation, etc.
 - Gameplay: Engaging and closely aligned with learning objectives
 - Knowledge delivery: Clear presentation of educational content within the game
 - Other: _____
8. What kind of art style would you prefer for the serious game? (You may select multiple options)
- Realistic style
 - Cartoon style
 - Pixel art style
 - Sci-fi style
 - Minimalist style
 - 3D style
 - 2D style
 - 2.5D style
 - Other: _____

Appendix C

System Functional Test Cases

Functional Test Case Lists

Test Case 1: User Registration

Objective:	Verify that the system supports user registration for a new account.
Input:	Enter a valid username, email, and password.
Expected Outcome:	The system successfully registers a new account and notifies the user of successful registration.
Result	Passed

Test Case 2: User Login

Objective:	Verify that the system supports user login with an existing account.
Input:	Enter the correct email and password.
Expected Outcome:	The system allows users to log in and enter the game world.
Result	Passed

Test Case 3: Display Background Introduction Before Entering the Game World

Objective:	Verify that the system provides a background introduction before entering the game world.
Input:	The user logs in and enters the game world.
Expected Outcome:	The system displays the background introduction.
Result	Passed

Test Case 4: Interacting with NPC to Obtain Quest Information

Objective:	Verify that the user can obtain quest information by interacting with NPCs.
Input:	The user interacts with an NPC in the game.
Expected Outcome:	The system displays quest information and updates the storyline.
Result	Passed

Test Case 5: Accepting a Quest from NPC

Objective:	Verify that the user can accept a quest from an NPC and start the challenge.
Input:	The user interacts with an NPC and accepts the quest.
Expected Outcome:	The system takes the user to the challenge page.
Result	Passed

More Test Cases on the Following Pages

Test Case 6: Rejecting a Quest from NPC

Objective:	Verify that the user can reject a quest from an NPC and later re-accept it.
Input:	The user interacts with an NPC and chooses to reject the quest.
Expected Outcome:	The quest is rejected, and the user can re-accept the quest.
Result	Passed

Test Case 7: Obtaining Information from Hint Items

Objective:	Verify that the user can obtain helpful information for completing challenges by interacting with hint items.
Input:	User clicks or views hint items.
Expected Outcome:	The system displays guidance information helpful for the challenge.
Result	Passed

Test Case 8: Scene Switching

Objective:	Verify that the system supports switching between different scenes in the game world.
Input:	The user interacts with the door, and selects enter.
Expected Outcome:	The system successfully switches to the target scene.
Result	Passed

Test Case 9: Viewing Leaderboard Information

Objective:	Verify that the system allows the user to view leaderboard information.
Input:	The user clicks the "SCORE BOARD" button to access the leaderboard page.
Expected Outcome:	The system displays scores and ranking information from the leaderboard.
Result	Passed

Test Case 10: Viewing Character Attributes

Objective:	Verify that the user can view their character's attributes and growth.
Input:	The user views the character attributes section on the top-left corner of the game world.
Expected Outcome:	The system displays character attribute information.
Result	Passed

Test Case 11: Viewing Challenge History

Objective:	Verify that the user can view their challenge history.
Input:	The user clicks "HISTORY" button to access the challenge history page.
Expected Outcome:	The system displays the user's challenge history.
Result	Passed

Test Case 12: Viewing User Help Manual

Objective:	Verify that the user can view gameplay guidance information.
Input:	The user clicks "HELP" button to access the help manual page.
Expected Outcome:	The system displays the user help manual.
Result	Passed

Test Case 13: Saving Game Progress and Exiting	
Objective:	Verify that the system allows the user to exit the game and resume progress upon return.
Input:	The user clicks "Quit" button to access to exit the game.
Expected Outcome:	The system saves the current progress, and resumes from the saved progress after the user logs back in.
Result	Passed

Test Case 14: Viewing Challenge Scene Description	
Objective:	Verify that the user can view the challenge scenario description.
Input:	The user enters the quest challenge page.
Expected Outcome:	The system displays the story background and description of the challenge scene.
Result	Passed

Test Case 15: Viewing Exploitation and Remediation Task Descriptions	
Objective:	Verify that the user can view the description of the exploitation and remediation tasks.
Input:	The user enters the quest challenge page.
Expected Outcome:	The system displays the task objectives and descriptions.
Result	Passed

Test Case 16: Accessing Vulnerable Web Application	
Objective:	Verify that the user can access the vulnerable web application.
Input:	The user clicks the button to access the vulnerable application.
Expected Outcome:	The system accesses the vulnerable application in a new tab of the default browser.
Result	Passed

Test Case 17: Resetting Vulnerable Web Application	
Objective:	Verify that the user can reset the vulnerable web application.
Input:	The user clicks the reset button.
Expected Outcome:	The system resets the vulnerable application to its initial state, and accesses the vulnerable application in a new tab of the default browser.
Result	Passed

Test Case 18: Submitting Exploitation and Remediation Task Answers	
Objective:	Verify that the user can submit answers for the exploitation and remediation tasks and receive feedback.
Input:	The user enters the answer and clicks the submit button/User modifies the code and clicks the update button.
Expected Outcome:	The system provides feedback on whether the task was successful or not.
Result	Passed

Test Case 19: Viewing Exploitation and Remediation Task Guidance	
Objective:	Verify that the user can view guidance for the exploitation and remediation tasks.
Input:	The user clicks "Guidance" selection to access the guidance page.
Expected Outcome:	The system displays the foundational knowledge required to complete the tasks.
Result	Passed

Test Case 20: Viewing Subtask Information and Hints

Objective:	Verify that the user can view subtask information and corresponding hints for the exploitation and remediation tasks.
Input:	The user clicks "Objectives"/"Hints" selection to access the relevant page.
Expected Outcome:	The system displays the subtask lists/corresponding hint information.
Result	Passed

Test Case 21: Viewing Exploitation and Remediation Solutions

Objective:	Verify that the user can view solutions for the exploitation and remediation tasks.
Input:	The user clicks "Solution" selection to access the solution page.
Expected Outcome:	The system displays the solutions for vulnerability exploitation/remediation after double confirmation.
Result	Passed

Test Case 22: Providing Remediation Task After Exploitation Task

Objective:	Verify that the system provides the corresponding remediation task after the exploitation task is completed.
Input:	The user completes the exploitation task.
Expected Outcome:	The system provides the remediation task section under the exploitation task section.
Result	Passed

Test Case 23: Modifying Vulnerable Application Code

Objective:	Verify that the user can modify the code of the vulnerable application to complete the remediation task.
Input:	The user uses the code editor to make modifications.
Expected Outcome:	The system allows the user to modify the code.
Result	Passed

Test Case 24: Submitting Remediation Task Answer

Objective:	Verify that the user can submit the remediation task answer and receive feedback.
Input:	The user modifies the remediation code and clicks the update button.
Expected Outcome:	The system provides feedback on whether the remediation task was successful or not.
Result	Passed

Test Case 25: Viewing Task Scoring Rules

Objective:	Verify that the user can view the scoring rules for the task.
Input:	The user clicks "Solution" selection to access the scoring rules page.
Expected Outcome:	The system displays the scoring criteria and requirements for the challenge.
Result	Passed

Test Case 26: Exiting Current Challenge

Objective:	Verify that the user can exit the current challenge at any time.
Input:	The user clicks the "Leave Challenge" button.
Expected Outcome:	The system display challenge has exited the page to ask the user back to the game application.
Result	Passed

Appendix D

Semi-structured Interview Questions

Usability Evaluation Interview Core Questions

1. What year are you in your undergraduate program?
2. Have you taken any cybersecurity or other computer-related courses?
3. How many levels of the Neon City Defender did you complete?
4. What did you like most about the Neon City Defender?
5. What did you like least about the Neon City Defender?
6. Would you recommend the Neon City Defender to your classmates?
7. Do you have any other opinions or suggestions regarding the Neon City Defender?
8. Why did you perform action X while using feature Y?
9. Which feature of the Neon City Defender do you think was most effective in helping you learn to identify, exploit, and remediate file upload vulnerabilities?

Appendix E

SUS Questionnaire

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

Figure E.1: SUS Questionnaire [12]

Appendix F

Pre-and Post-tests Questions for Teaching Effectiveness Evaluation

1. What is the primary risk associated with arbitrary file uploads in web applications?
 - A) Slow performance
 - B) High resource usage
 - C) Remote command execution
 - D) Increased storage costs
2. Which scripting environment is commonly used as a web backdoor for command execution?
 - A) Web Shell
 - B) JavaScript Engine
 - C) HTML5 Canvas
 - D) CSS Preprocessor
3. Which type of file is most commonly used to perform file upload attacks?
 - A) Image files
 - B) Text files
 - C) Executable files
 - D) Audio files

4. Which of the following techniques can be used to bypass file upload extension validation?
 - A) File content encryption
 - B) MIME type spoofing
 - C) Uploading large files in segments
 - D) Modifying file metadata
5. Which HTTP header can be used to disguise the real type of an uploaded file?
 - A) User-Agent
 - B) Referer
 - C) Content-Type
 - D) Accept-Encoding
6. Which method can prevent direct access to uploaded files?
 - A) Storing files in a public directory
 - B) Setting specific permissions for uploaded files
 - C) Using .htaccess files to deny access
 - D) Allowing all users to access uploaded files
7. For secure handling of file uploads, where should the uploaded files be stored?
 - A) In the web root directory
 - B) On an external storage service
 - C) In a directory isolated from the web root
 - D) Directly in the database
8. In which situation should whitelisting be preferred over blacklisting for managing file uploads?
 - A) When the types of files are diverse and constantly changing
 - B) When known dangerous file types are few
 - C) When reducing the complexity of security policy management is desired
 - D) When file uploading is not a critical security focus

9. When designing a secure file upload feature, which of the following measures is not recommended?
- A) Executing files immediately after upload to verify their content
 - B) Implementing server-side file type detection and content verification
 - C) Prohibiting direct access to any uploaded files
 - D) Ensuring no scripts are executed in the file upload directory

Appendix G

Gameplay UI Implementation

Screenshots

The screenshot shows a registration form titled "Register" on a dark purple background. The form includes a "Back To Login" button in the top right corner. Below the title, there are four input fields: "Username", "E-mail", "Password", and "Confirm Password". Each field has a placeholder text: "Please enter Username here", "Please enter E-mail here", "Please enter Password here", and "Please enter Confirm Password here". At the bottom of the form is a large red "REGISTER" button. The browser window title is "CyberpunkNeonCityDefender".

Register

Back To Login

Username

Please enter Username here

E-mail

Please enter E-mail here

Password

Please enter Password here

Confirm Password

Please enter Confirm Password here

REGISTER

Figure G.1: Registration Page



Figure G.2: Tutorial for Moving in the Game World



Figure G.3: Interaction with NPC in Level 2 Scene



Figure G.4: Exploring the Scene in Level 3

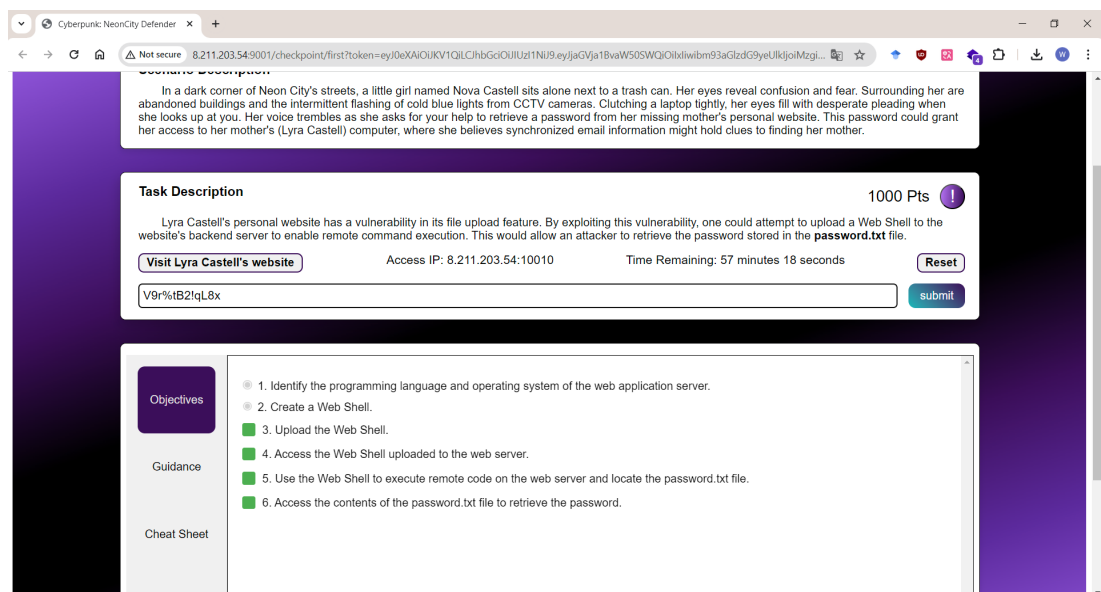


Figure G.5: Subtask Recognition during Vulnerability Exploitation in Level 1 Challenge

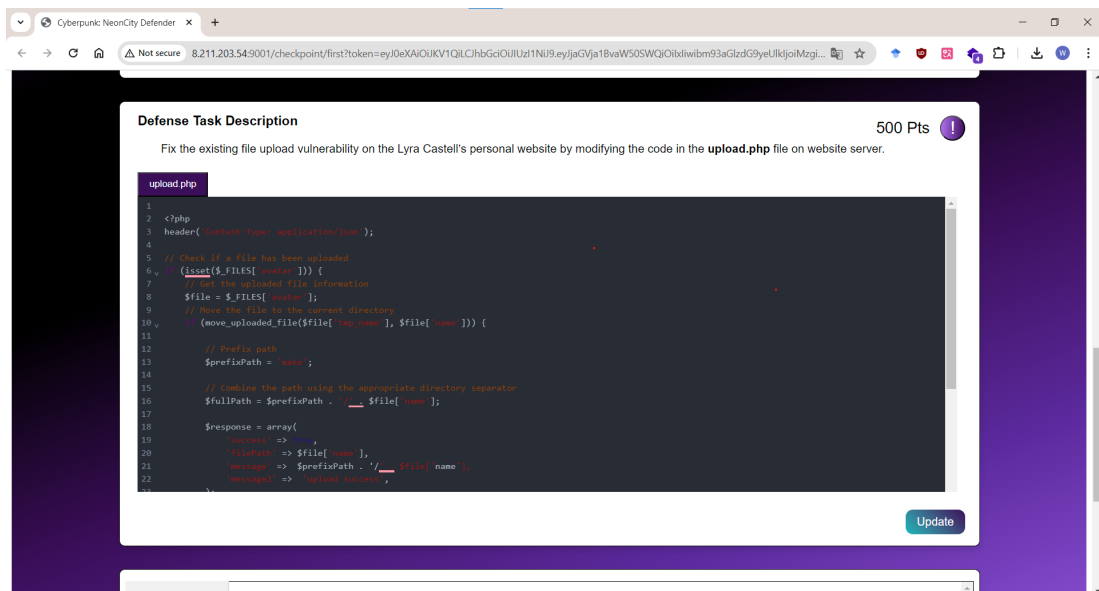


Figure G.6: Code Editor for Vulnerability Fix Task in Level 1 Challenge

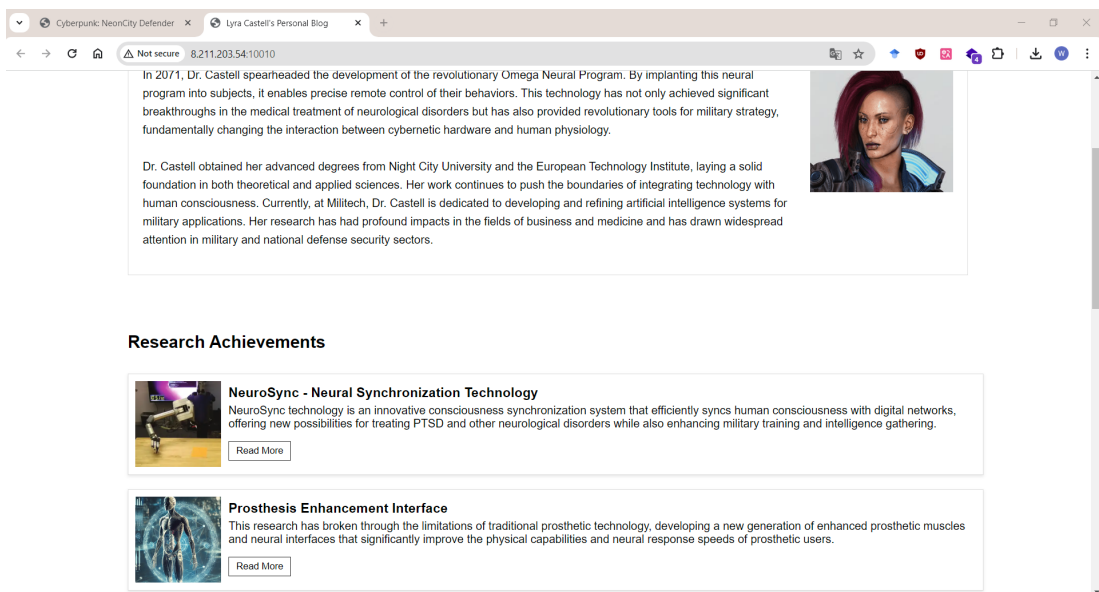


Figure G.7: Vulnerable Application of Level 1 Challenge

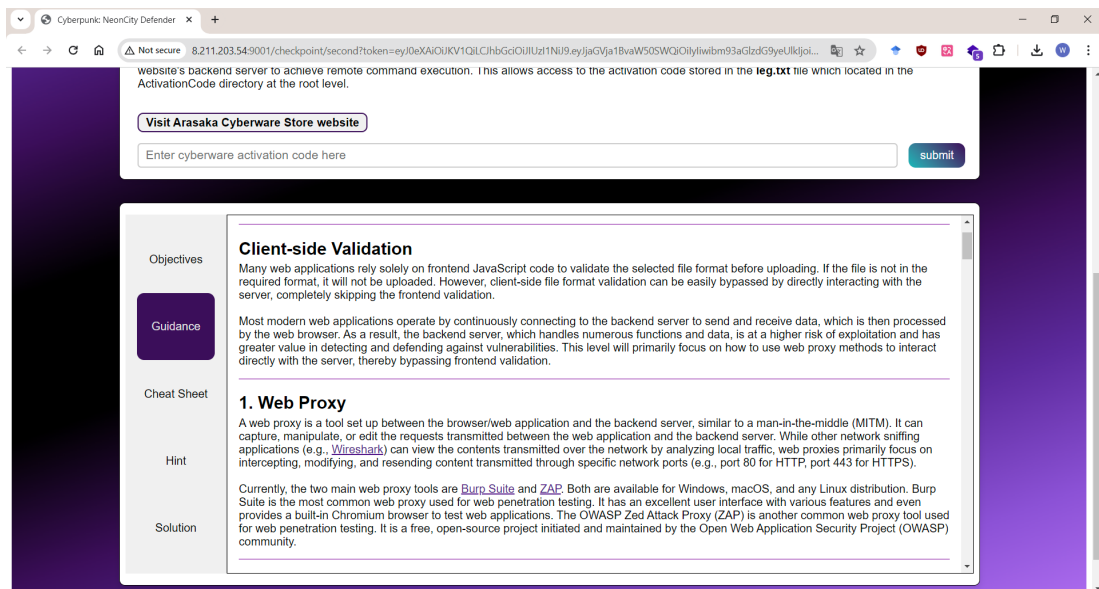


Figure G.8: Guidance for Vulnerability Exploitation in Level 2 Challenge

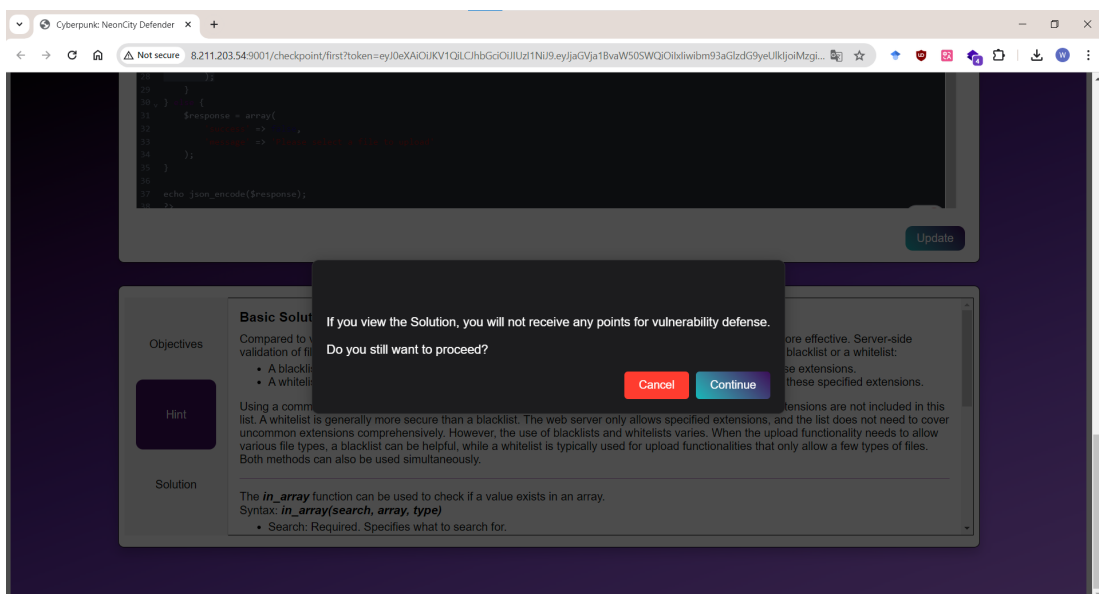


Figure G.9: Double Confirmation Popup for Viewing Solution in Level 2 Challenge

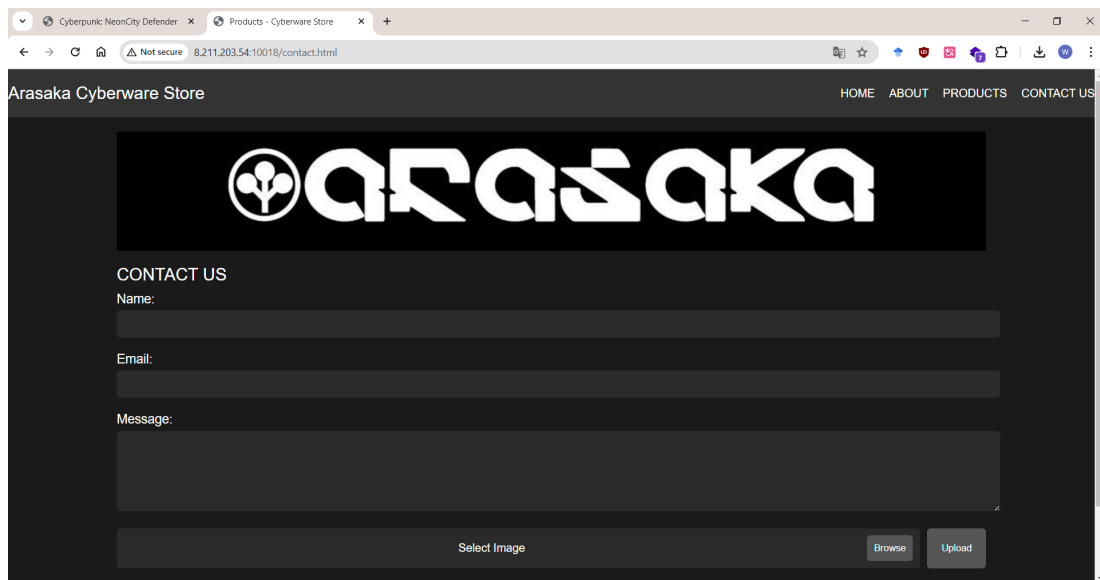


Figure G.10: File Upload Page of Vulnerable Application in Level 2 Challenge

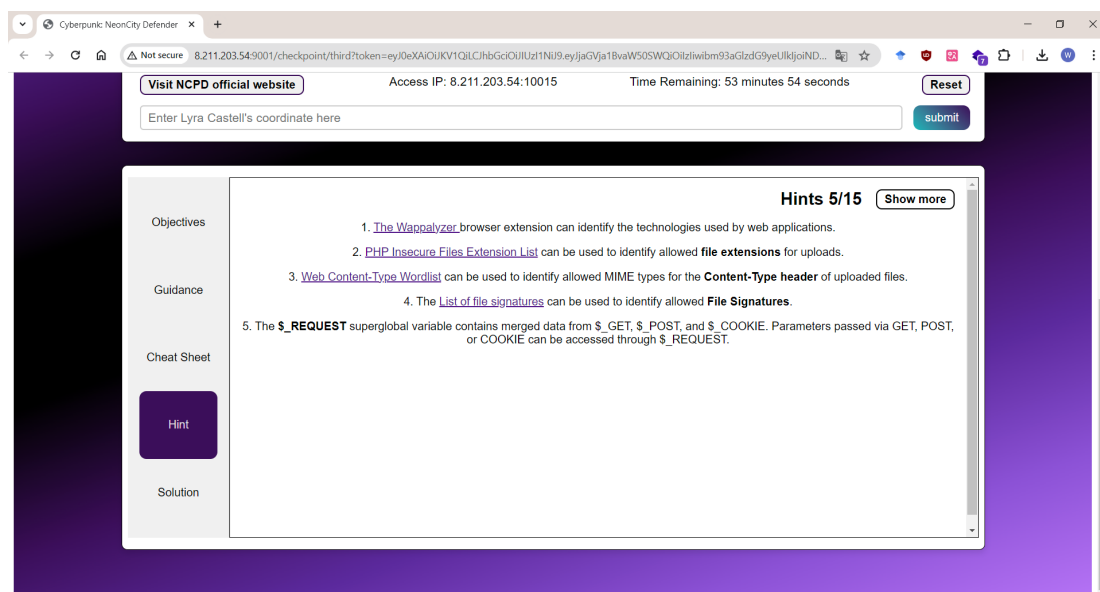


Figure G.11: Hint Page for Vulnerability Exploitation in Level 3 Challenge

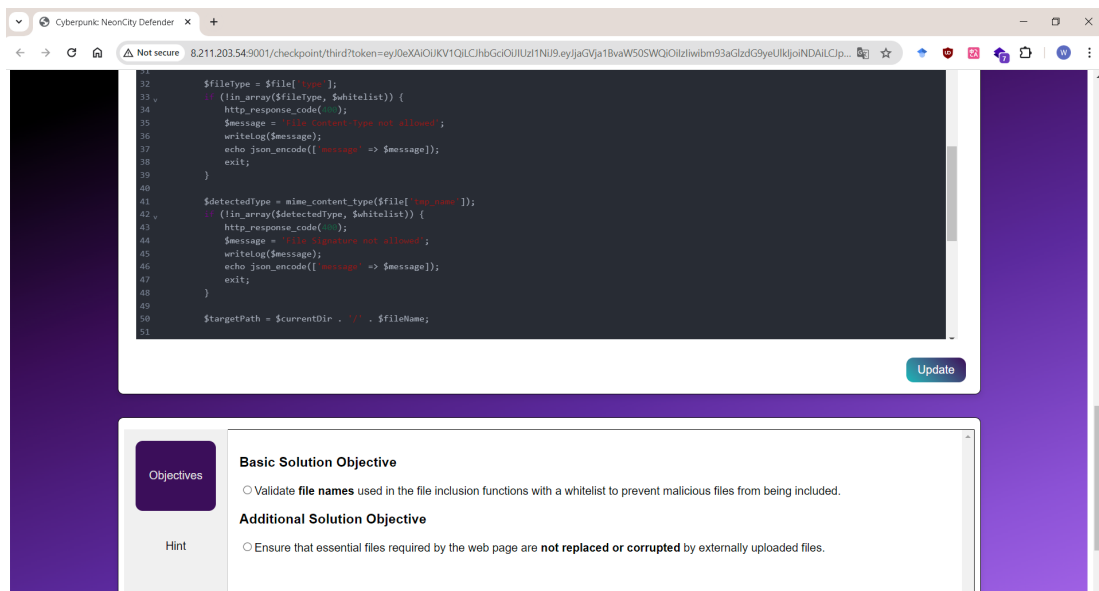


Figure G.12: Subtask Recognition for Vulnerability Fixing in Level 3 Challenge

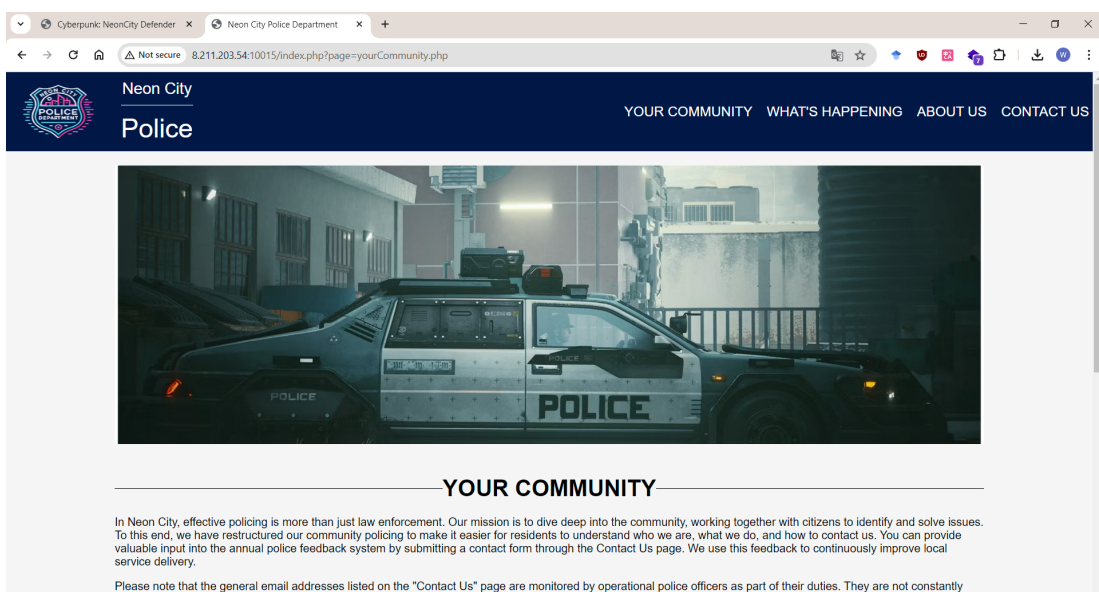


Figure G.13: Homepage of Vulnerable Application in Level 3 Challenge

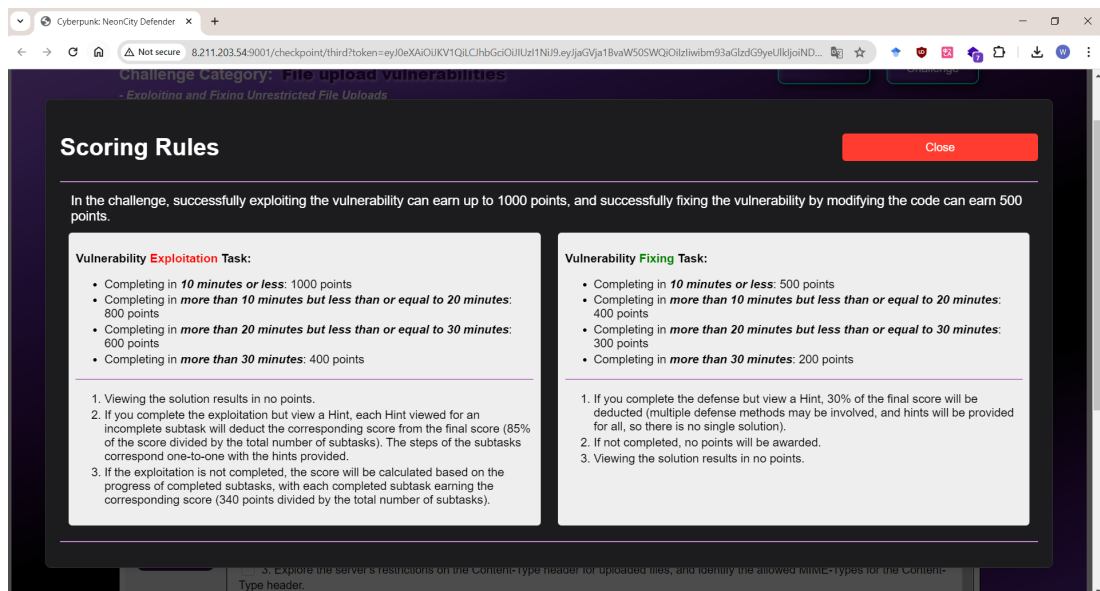


Figure G.14: Scoring Rules Page for Challenges

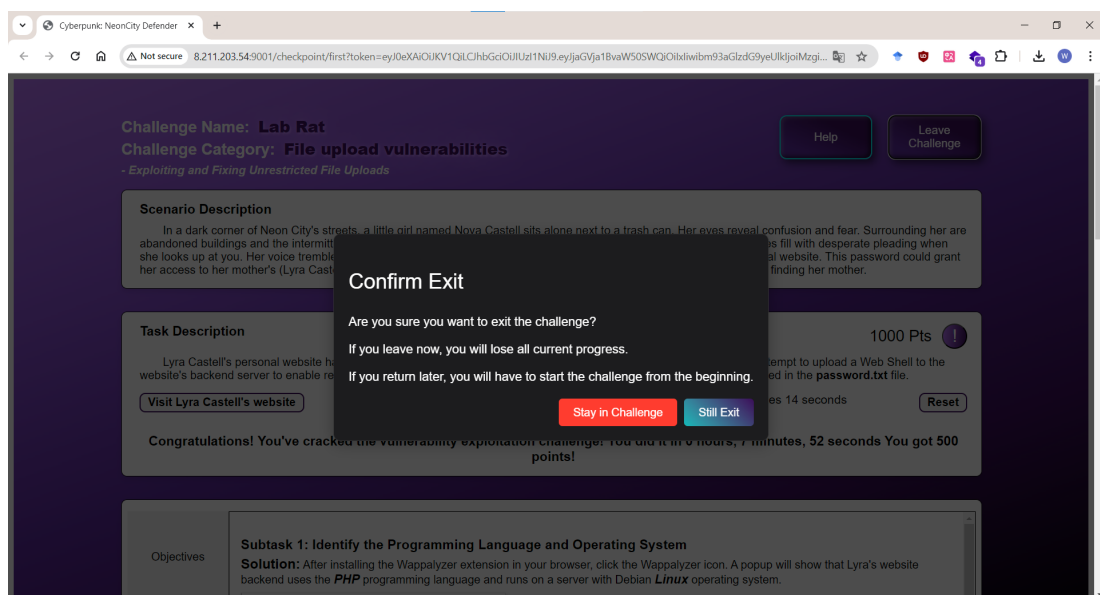


Figure G.15: Double Confirmation for Exiting Challenge

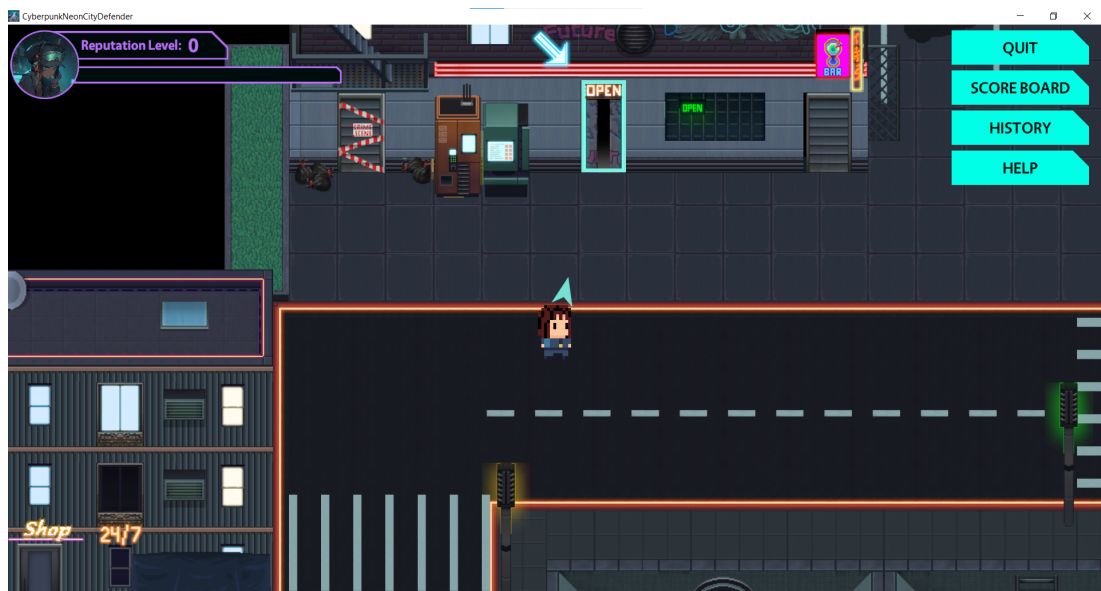


Figure G.16: Inter-level Navigation Function in the Game World

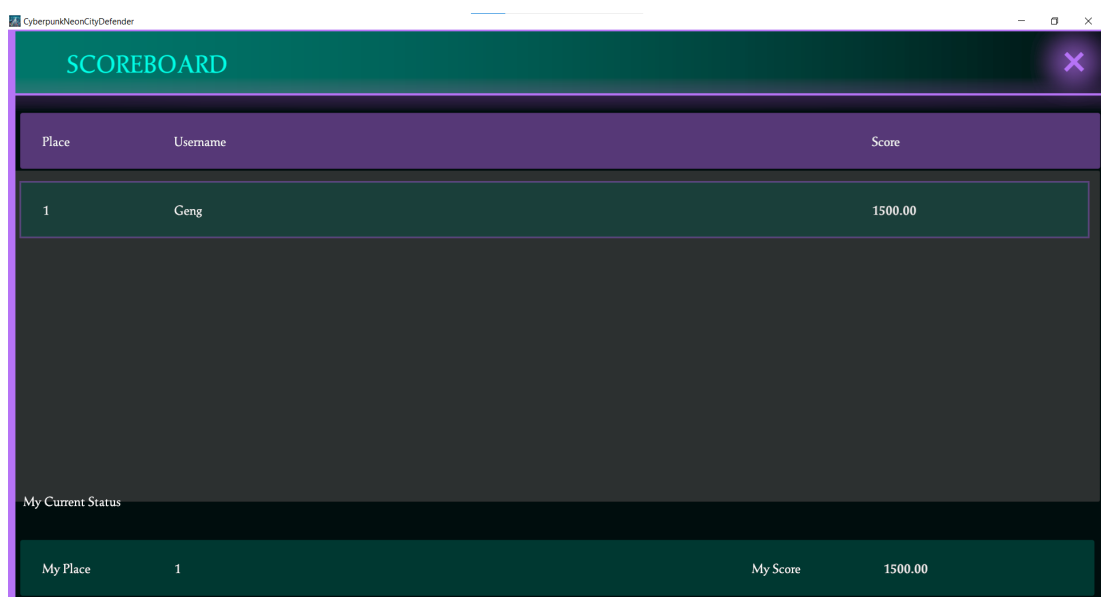
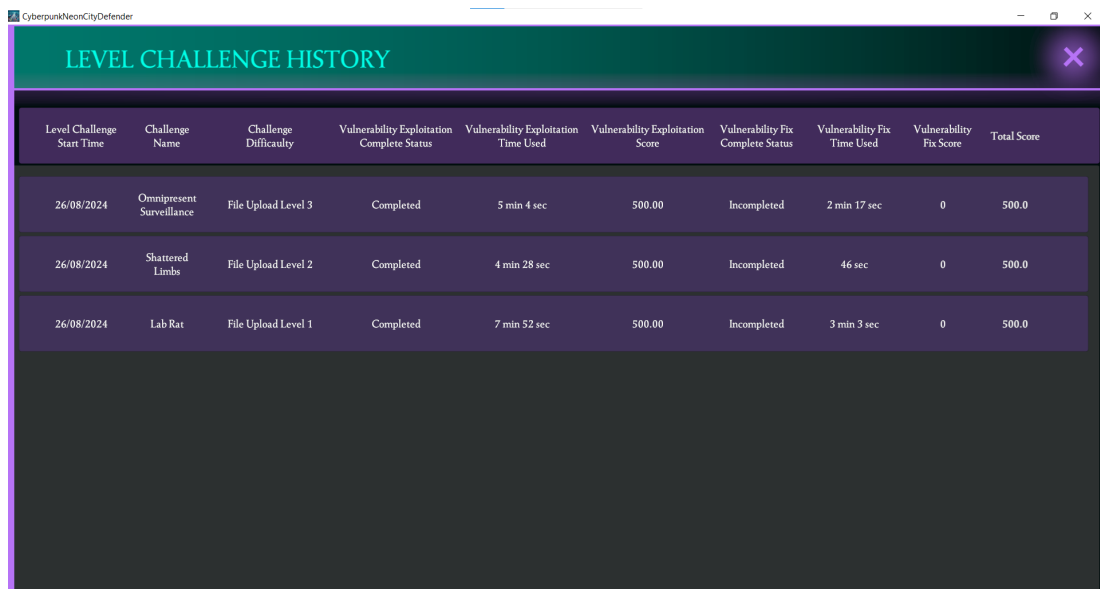
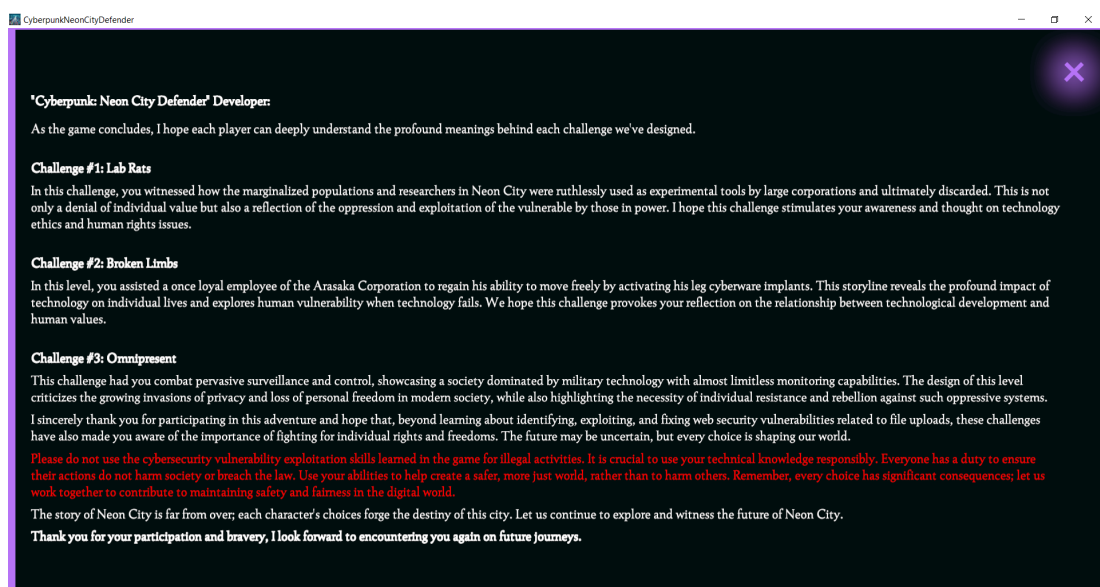


Figure G.17: Leaderboard Page



Level Challenge Start Time	Challenge Name	Challenge Difficulty	Vulnerability Exploitation Complete Status	Vulnerability Exploitation Time Used	Vulnerability Exploitation Score	Vulnerability Fix Complete Status	Vulnerability Fix Time Used	Vulnerability Fix Score	Total Score
26/08/2024	Omnipresent Surveillance	File Upload Level 3	Completed	5 min 4 sec	500.00	Incompleted	2 min 17 sec	0	500.0
26/08/2024	Shattered Limbs	File Upload Level 2	Completed	4 min 28 sec	500.00	Incompleted	46 sec	0	500.0
26/08/2024	Lab Rat	File Upload Level 1	Completed	7 min 52 sec	500.00	Incompleted	3 min 3 sec	0	500.0

Figure G.18: Challenge History Page



'Cyberpunk: Neon City Defender' Developer:

As the game concludes, I hope each player can deeply understand the profound meanings behind each challenge we've designed.

Challenge #1: Lab Rats

In this challenge, you witnessed how the marginalized populations and researchers in Neon City were ruthlessly used as experimental tools by large corporations and ultimately discarded. This is not only a denial of individual value but also a reflection of the oppression and exploitation of the vulnerable by those in power. I hope this challenge stimulates your awareness and thought on technology ethics and human rights issues.

Challenge #2: Broken Limbs

In this level, you assisted a once loyal employee of the Arasaka Corporation to regain his ability to move freely by activating his leg cyberware implants. This storyline reveals the profound impact of technology on individual lives and explores human vulnerability when technology fails. We hope this challenge provokes your reflection on the relationship between technological development and human values.

Challenge #3: Omnipresent

This challenge had you combat pervasive surveillance and control, showcasing a society dominated by military technology with almost limitless monitoring capabilities. The design of this level criticizes the growing invasions of privacy and loss of personal freedom in modern society, while also highlighting the necessity of individual resistance and rebellion against such oppressive systems. I sincerely thank you for participating in this adventure and hope that, beyond learning about identifying, exploiting, and fixing web security vulnerabilities related to file uploads, these challenges have also made you aware of the importance of fighting for individual rights and freedoms. The future may be uncertain, but every choice is shaping our world.

Please do not use the cybersecurity vulnerability exploitation skills learned in the game for illegal activities. It is crucial to use your technical knowledge responsibly. Everyone has a duty to ensure their actions do not harm society or breach the law. Use your abilities to help create a safer, more just world, rather than to harm others. Remember, every choice has significant consequences; let us work together to contribute to maintaining safety and fairness in the digital world.

The story of Neon City is far from over; each character's choices forge the destiny of this city. Let us continue to explore and witness the future of Neon City.

Thank you for your participation and bravery, I look forward to encountering you again on future journeys.

Figure G.19: Game Ending Screen with Reminder to Use learnt Knowledge Responsibly

Appendix H

Participants' information sheet

Project title:	Tool for teaching web application exploits and defences
Principal investigator:	Dr Myrto Arapinis
Researcher collecting data:	Wenjia Geng
Funder (if applicable):	None

This study was certified according to the Informatics Research Ethics Process reference number 945616. Please take time to read the following information carefully. You should keep this page for your records.

Who are the Researchers?

Dr Myrto Arapinis and Wenjia Geng.

Purpose of the Study

The purpose of the study is to design and develop a CTF-style serious game as a web application security teaching tool aimed at enhancing students' theoretical knowledge and practical skills in identifying, exploiting, and defending against web application vulnerabilities. The project seeks to improve the learning process by addressing the challenges of traditional educational approaches and the high entry barrier of existing CTF competitions through the engaging and educational format of serious games.

Why Have I Been Asked to Take Part?

You have been invited to participate in this study because you are among our target group. The project will mainly target undergraduate and graduate students who are beginners or already possess some knowledge of web application security. Your participation will help us assess the effectiveness of this serious game as a tool for teaching how to identify, exploit, and defend against web vulnerabilities for target users from diverse backgrounds.

Do I Have to Take Part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time up until 12/August/2024 without giving a reason. After this point, personal data will be deleted and anonymised data will be combined such that it is impossible to remove individual information from the analysis. Your rights will not be affected. If you wish to withdraw, contact the PI. We will keep copies of your original consent and of your withdrawal request.

What Will Happen If I Decide to Take Part?

If you decide to participate in this study, you will help us to evaluate the serious game's functionality and usability, and then its effectiveness in improving teaching outcomes.

1. Kinds of Data Being Collected:

- Basic demographic data including education level and background in coding/web security.
- Pre- and Post-Test scores and details to assess the teaching effectiveness of the game. Participants will complete tests on vulnerability exploitation and defense knowledge before and after playing the game.
- Game interaction data on how participants interact with the game, including completion of specific tasks, choices made within the game, and engagement with different elements of the game.

- Interview responses from semi-structured interviews after the gameplay experience, including participants' experiences, feelings, and feedback on the game's usability and educational impact.

2. Means of Collection:

The basic demographic data will be collected through the information provided during the game character creation process within the game. Pre- and post-test scores will be gathered using standardized tests on vulnerability exploitation and defense knowledge conducted before and after gameplay. The serious game system will automatically collect data on the frequency and duration of participants' interactions with game elements, as well as track the progress of challenge task completion. After the post-test, semi-structured interviews will be conducted to collect qualitative data on participants' experiences, feelings, and feedback concerning the game's usability and teaching improvement effectiveness. The interview may be recorded in audio format with consent to ensure accurate capture and analysis of feedback data.

3. Duration of Session:

Each session, including gameplay, tests, and the post-game interview, is expected to last approximately 2.5 hours.

4. How Often, Where, and When:

The gameplay, pre- and post-game tests, and the semi-structured interview conducted after the game will each take place once per participant individually and are intended to be completed sequentially during a single meeting. The meeting will be scheduled to occur at any time convenient for the participant between late July and early August 2024. All sessions will be conducted in person or online in a manner that complies with the research confidentiality requirements.

Are There Any Risks Associated with Taking Part?

There are no significant risks associated with participation.

Are There Any Benefits Associated with Taking Part?

There are no benefits associated with participation.

What Will Happen to the Results of This Study?

The results of this study may be summarised in published articles, reports, and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of four years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data Protection and Confidentiality:

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher, including Dr Myrto Arapinis and Wenjia Geng. All electronic data will be stored on a password-protected, encrypted computer on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or SharePoint). All paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses to minimize risk.

What Are My Data Protection Rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance with Data Protection Law. You also have other rights, including rights of correction, erasure, and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments, and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who Can I Contact?

If you have any further questions about the study, please contact the lead researcher, Wenjia Geng, at s2494477@ed.ac.uk. If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated Information:

If the research project changes in any way, an updated Participant Information Sheet will be made available on <http://web.inf.ed.ac.uk/infweb/research/study-updates>.

Alternative Formats:

To request this document in an alternative format, such as large print or on coloured paper, please contact Wenjia Geng at s2494477@ed.ac.uk.

General Information:

For general information about how we use your data, go to edin.ac/privacy-research.

Appendix I

Participants' consent form

Project title:	Tool for teaching web application exploits and defences
Principal investigator:	Dr Myrto Arapinis
Researcher collecting data:	Wenjia Geng
Funder (if applicable):	None

By participating in the study you agree that:

1. I have read and understood the Participant Information Sheet for the above study, had the opportunity to ask questions, and any questions I had were answered to my satisfaction.
2. My participation is voluntary, and I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights.
3. I consent to my anonymised data being used in academic publications and presentations.
4. I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.

Please tick yes or no for each of these statements.

- | | | |
|--|------------|-----------|
| 1. I agree to being audio recorded. | Yes | No |
| 2. I allow my data to be used in future ethically approved research. | Yes | No |
| 3. I agree to take part in this study. | Yes | No |

Name of person giving consent

Signature

dd/mm/yy

Name of person taking consent

Signature

dd/mm/yy

Participant number: _____