TU WIEN Informatics

Implementing an Educational Computer Security Game Targeted at Computer Science Students

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Media and Human-Centered Computing

eingereicht von

Julia Grill, BSc
Matrikelnummer 01501940

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Purgathofer
Mitwirkung: Univ.Ass. Dipl.-Ing. Maximilian Robert Ulreich, BSc

Wien, 4. Juli 2023

Julia Grill	Peter Purgathofer

TU WIEN Informatics

Implementing an Educational Computer Security Game Targeted at Computer Science Students

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Media and Human-Centered Computing

by

Julia Grill, BSc
Registration Number 01501940

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Purgathofer
Assistance: Univ.Ass. Dipl.-Ing. Maximilian Robert Ulreich, BSc

Vienna, 4th July, 2023

Julia Grill | Peter Purgathofer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Julia Grill, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 4. Juli 2023

Julia Grill

Acknowledgements

I would like to thank my supervisor Prof. Peter Purgathofer and his assistant Maximilian Robert Ulreich for their support throughout my thesis. Their feedback helped shape the outcome of the game. I would also like to thank the Security and Privacy Group at TU Vienna for insightful discussions about educational games. Special thanks to Prof. Edgar Weippl for taking the time to discuss gamficiation in computer security courses. Lastly, I would like to thank my partner Stephanie for her continuous support and feedback on the game. This thesis would not have been possible without her.

Kurzfassung

Lernspiele sind ein sehr effektiver Ansatz, unterschiedliche Themen zu vermitteln. Computer Security kann für Informatikstudierende ohne Vorkenntnisse ein entmutigendes Thema sein. Im Rahmen dieser Arbeit wurde ein Spielprototyp entwickelt, der Spieler und Spielerinnen Computer Security Themen näherbringt. Die Effektivität des Spiels wurde mithilfe von Umfragen und Spieltests evaluiert. Die Arbeit hatte als Ziele Veränderungen von Spieler- und Spielerinneninteresse nachzuvollziehen, Spieler- und Spielerinnenwissen zu erhöhen und Probleme im Zusammenhang mit Game Design zu verstehen. Basierend auf den Ergebnissen der Evaluierungen kommt die Autorin zu dem Schluss, dass Spieler und Spielerinnen Wissen gewonnen haben und das Spieler- und Spielerinneninteresse in diesem Gebiet gestiegen ist. Es wurden mehrere Game Design Probleme festgestellt, die in zukünftigen Iterationen des Spiels behoben werden sollten.

Abstract

Educational games are a very effective approach to teaching various topics. Computer security can be a daunting topic for computer science students with no background in the field. In the context of this thesis a game prototype was developed which teaches players about computer security topics. To evaluate the effectiveness of the game, questionnaires were created and playtesting sessions were held. The aims of this thesis were to understand any changes in player interest, increase in player knowledge in the field of computer security, and to understand any issues related to game design. Based on the results of the evaluations, the author concludes that players gained knowledge from playing the game and player interest in the field increased. Several game design issues were detected, which should be addressed in future iterations of the game.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Problem Statement

In the world of Smart Homes and Internet of Things understanding the importance of security and how to counterattack security threats is a vital skill for everyone. First semester computer science students might be enthusiastic about technology but have yet to enrol in any security related courses. Therefore, it is beneficial to confront them with the topic of security as early as possible in their studies.

The mandatory first semester course *Denkweisen der Informatik* [1] introduces students to many different aspects of Computer Science - including security - in the form of slides and (text) challenges. It is using a gamified approach to teaching by utilising the custom platform *Aurora* [2], which acts similarly to a Massive Open Online Course (MOOC). On *Aurora* students can exchange thoughts and directly upload their assignments. The platform consists of different *Thinking labs* which each focus on a specific Computer Science topic.

Despite the well-known benefits of educational games on students' learning rates [3], there is a significant lack of such games on the platform. The aim of this work is to develop an educational game which can be utilised in the lab *Criminal Thinking* to increase interest in the field of security as well as facilitate the learning process.

Taking the aforementioned points into consideration, the thesis aims to answer the following questions:

- What effect does an educational game have on the interest in the field of security in computer science students?
- To what extent can a security-focused game convey the basics of security attacks in an interesting way?

- How effective is the game in teaching players the basics of security attacks?

1.2 Expected Results

The aim of this work is to develop a web-based serious game which teaches computer science students about common security attacks in an engaging manner. The initial game design will be based on the author's original idea and expert feedback. The content of the game will focus on security topics covered in the course *Denkweisen der Informatik* [1]. The prototype will be played and reviewed mostly by students but also by people with a wide range of computer security knowledge. The results of the reviews will determine the effectiveness of the prototype as well as hint at future changes.

In order to gather player impressions, two questionnaires will be used. One questionnaire will be answered pre-play, one will be answered post-play. This way, change in interest and knowledge can be easily compared. Additionally, the feedback provided in the questionnaire measures the effectiveness of the game and highlights future improvements to the prototype. In addition to questionnaires, playtesting will be utilised to see understand how players interact with the game and detect any design flaws. The game should give players an overview of important security topics and should increase their interest in the field.

CHAPTER 2

Methods

The game was developed and evaluated utilising different approaches. First, input was gathered from expert, then the game was implemented and finally the game was evaluated by both experts and players.

2.1 Literature Review

A literature review was conducted to understand the effectiveness of security focused serious games on knowledge retention. The research focused on keywords: *Serious Security Games*, *Educational Gaming* and *Gamification for Higher Education*. The motivation behind this methodology was to gain an understanding of the status quo of educative security games in higher education. Notably, the aim was to learn about common trends and issues. Papers that were published post-2015 with ≥ 10 citations were prioritised.

2.2 Expert Interview

An expert interview was conducted with Professor Edgar Weippl. He is a professor for security at the University of Vienna and has lead many first semester security courses [4]. The intention behind this method was to gain insights on how computer security is taught and utilise these aspects in the game. In particular, the aim was to understand which information the game should convey, which aspects the game should focus on and how to approach gamification in education. The interview was conducted via Zoom.

2.3 Focus Group

A focus group session was held with the security and Privacy group at TU Vienna prior to starting with the game development process [5]. The motivation behind this methodology

was to brainstorm possible game ideas and game content with experts which could later be implemented in the game. The group has a lot of experience with developing gamified security teaching solutions. Similar to the expert interview, the aim was to gather input and feedback on game content and game design. The focus group session was held in-person.

2.4 Game Design & Development

2.4.1 Game Scope & Design

The game design was heavily influenced by the results of the focus group. The game scope was influenced by the content of the course *Denkweisen der Informatik* [1].

The author came up with the initial game design idea. The idea was developed further based on feedback received during the focus group session. Then, it was iteratively improved upon based on feedback from the supervisor.

2.4.2 Game Implementation

The game was implemented using the Open Source game development engine *Godot* [6]. Assets were mostly self-designed using the software Aseprite [7]. Two assets were taken from *itchio.io* [8].

2.5 Collection & Evaluation of Feedback

To evaluate the game, both expert and player feedback was gathered and analysed. Focus was put on player feedback. Player feedback was gathered in the form of questionnaires and play-testing. The expert feedback was used to understand the bigger context of game-design issues.

2.5.1 Expert Feedback

Expert feedback was gathered in form of a game-presentation. The motivation behind this presentation was to gather expert reactions to detect major issues in the game. The game presentation consisted of covering the background of the work and a presentation of the game-play. It was held in front of the participants of the focus group. Feedback was given after the presentation, followed by a discussion covering the general approach to educational games. The points from the feedback and discussion were gathered and summarised.

2.5.2 Player Questionnaire Feedback

To evaluate the game from the player's perspective in a quantitative manner, two questionnaires were created that players had to fill out. The questionnaires offered

straight-forward player feedback and analysis of player impressions which are key elements when measuring the game's effectiveness. They were created using Cryptpad [9]. One questionnaire was answered pre-play, the other was answered post-play. The questions consisted of multiple choice and free-text form questions. The free-text form questions were not mandatory to submit the questionnaire. Players could only access the post-game questionnaire once they have finished the main game.

Collection of Feedback

Participants needed to be enrolled students to qualify for participating in the questionnaire. Participants were reached through forum posts in the courses *Introduction to Security* and *Denkweisen der Informatik* promoting the game. They were also reached in-person in the course *Gameful Design*. The questionnaires were shared by the respective lecturers. Since *Introduction to Security* is a third semester course and *Denkweisen der Informatik* is a first semester course, it can be assumed that participants were new to the field of security. *Gameful Design* is a course offered in the master curriculum of Media- and Human-Centered Computing. This master requires no knowledge in the field of security. The target number of participants was ten to guarantee relevant evaluation results. The pre-game questionnaire had 34 participants while the post-game questionnaire had twelve participants.

Evaluation of Feedback

The focus of the evaluation was to understand and compare player security knowledge and interest pre- and post-play. For this, relevant security questions were asked pre- and post-play. Focus was put in understanding changes in interest and knowledge.

2.5.3 Playtesting Feedback

Playtesting sessions were conducted with several people to understand how players interact with the game and to understand their thinking process. Similar to the player questionnaire, the aim of the playtesting sessions was to evaluate the game from a player's perspective in a qualitative manner. As the playtesting sessions were held in-person, it enabled the author to directly interact with the testers. The questions were similar to the questionnaire in that they focused on understanding player knowledge, game-play and general game feedback.

Collection of Feedback

Playtesters were reached with the help of the supervisor. Their computer security knowledge ranged from beginner to computer security tutor and IT didactic professor. The number of playtesters was six.

Evaluation of Feedback

The focus in the evaluation was to understand how players interact with the game, meaning they complete tasks and mini-games as well as see their immediate reactions to game content. This also enabled easy detection of potential issues. Additionally, discussions with and direct feedback from playtesters were included in the evaluation.

CHAPTER 3

State of the Art

It has been confirmed in multiple studies that gaming can lead to more information being retained, especially due to the enjoyment of the activity [3], [10]. It also leads to higher engagement of the students [11]. The studies covering gamification of security aspects for teaching students and working professionals covers a wide range of approaches: from computer games, to board- and card games to gamified learning platforms.

Vykopal and Barták (2016) [12] have conducted research on teaching security principles through a serious game. In particular, focus was put on understanding how players interact with the game and what can be learnt for developing security training. The game was playtested by computer Science undergraduate and PhD students. However, the authors pointed out unbalanced game design which in return limits the results of the studies. Boopathi et al. (2015) [13] also confirm the effectiveness of gamification of key security components for students. They conducted a Capture the Flag (CTF) contest where students had to complete multiple assignments covering different security aspects. It should be noted that their research did not discuss the longevity of the learning effect. Similar results were presented by Beltrán et al. (2018) [14].

Schreuders and Butterfield (2016) [15] covered the effects of teaching computer security in a gamified setting at university. The authors implemented their own learning platform which used “quest rewards“ and “XP“ to measure performance. Whilst student engagement was high, it led to a lot of overhead for the tutors and professors. Additionally, the results did not conclude whether interest in the topic was increased. However, students noted that they enjoyed the gamified approach to teaching.

Roepke and Schroeder (2019) [16] point out that in their review of security games that many games are missing relevance for their content and are thus not teaching sustainable knowledge. They also noted that most of their reviewed games were not targeted at computer Science students. It is also mentioned that many security games do not survive the prototype phase due to most games being developed in the context of specific research.

Yasin et al. (2019) [17] developed a serious board game which aimed at improving software security awareness for players with minimum knowledge of security. In the game the players are split into teams and are tasked with defending a hospital from potential ransomware attacks. The players' main aims are to evaluate possible vulnerabilities and uncover the people involved in the attacks. This is done by completing challenges the board presents to the players. The game builds on communication amongst team members where each player takes over a specific roll. They need to correctly identify and judge threats in order to successfully complete the game. For the evaluation of the game playtesting sessions with students were conducted. Players stated that they understood the security concepts presented in the game. They were also able to successfully develop security attacking strategies themselves.

Hart et al. (2020) [18] presented a serious card game for raising security awareness targeted at working professionals with no professional background in security. The game conveys a wide array of attacks and defences to players. It utilises a board which highlights the context of game's assets for players to properly identify risks. A game master is required to explain background information. Each turn one player is an attacker and all other players are defenders. The attacking player draws an attacking card and every defending player has to select a defending card. The game master then explains to the defenders why or why not a certain card would work. Realistic threats from the attacker are rewarded with points. The results of the game evaluation were very positive. Players could pick up the game easily and they saw their knowledge increase.

Frey et al. (2019) [19] presented a serious board game about physical and computer security awareness for organisations. The game board consists of a LEGO board depicting some kind of physical infrastructure, e.g. network cables and employee supervision. Players need to defend their systems from different kinds of attacks. Every turn players can invest in upgrading their defences which is done by adding LEGO pieces to the board. The game master then conducts attacks against the players' infrastructure. The game was playtested by experts and individuals with no computer security experience. The groups were divided by player knowledge. The playtesting yielded interesting results. The experts teams went into the game with tunnel vision, primarily focusing on tasks that they deemed the most important. This left them vulnerable to other attacks. The groups with no security knowledge thrived through information gathering. The results of the evaluation showed that especially players with little to no security knowledge gained the most from playing the game. They saw very high value in playing the game.

Mostafa and Faragallah (2019) [20] developed six small educational security games each covering a different genre and security topic. The authors' aim was to understand which factors influence the effectiveness of an educational security game. In particular they were interested in which game genre is the most effective for the given context. They separated the games into two categories: loosely connected and highly integrated content. In the former the content has no effect on game design whereas in the latter game design revolves around the topic. The evaluation of the games was done with tests and questionnaires. The players were undergraduate computer science students. Focus was

put on understanding knowledge gain, enjoyment, complexity, and ease of use. Games with engaging stories and highly integrated content saw better evaluation results. Players took away the most from playing these games. The game covering web security featuring the genre *Action/Adventure* was deemed the most effective due to its rich story. However, it should be noted that the authors mention that the most limiting factor in educational games is the game design itself.

To summarise, peer-reviewed literature covering the benefits of gamified approaches to teaching security aspects is promising. Most of them yield very positive educational benefits. A very effective approach appears to be to expose players to both attacking and defending situations to understand the complexity of security concerns. Ideally, they should also have to justify and discuss their selected options [18], [19]. Another approach is to engage players in a rich story relevant to a respective security topic [20]. It should be noted that game design is the most limiting factor in the effectiveness of an educational game. A generalised statement on which topic and game design combination is the most effective cannot be made. Several studies highlight the high knowledge retention rates of security games when they are highly engaging [17]–[20]. All of these studies show promising results for future developments given the high game design standards are met. Unfortunately, in many cases game design feedback does not lead to further improvement of the game due to prototypes not being further developed [16].

3.1 Practical Examples

There are some practical examples of educational security games. In addition to games, there are many examples of gamified security challenges which aim to teach participants about core security talents in a protected environment, e.g. Tryhackme [21]. Whilst these challenges can be very beneficial for players, they will not be covered in this section as they are not games in the classical sense. Still, the author wanted to acknowledge the existence of these challenges.

For some games, the educational benefit is the main focus point, for others it is a side-effect of playing the game. An example of a game focusing on the educational aspect of computer security is the game *ThreatGEN: Red vs. Blue* [22]. In this game, players either take the role of hackers or defenders. The goal is to either infiltrate or defend a system. It utilises turns and a turn time limit. The game features network views with connected devices. Different actions can be bought by either weakening or defending the system. It holds a *Very Positive* rating on the gaming platform STEAM.

One of the first successful educational computer security games was *Uplink* released in 2006 [23]. In this game the player takes over the role of an agent tasked with hacking various entities. The player can accept different missions. The mission description contains the mission goal and a vague description of how to achieve mentioned goal. As an example mission, the player has to gain access to a test machine by breaking a security layer and by accessing a specific file without leaving a trace. The player's system can be

improved by buying upgrades, i.e. password breakers. The game gets increasingly more complex as the game progresses. It holds a *Very Positive* rating on STEAM.

A Normal Lost Phone showcases the dangerous world of Social Engineering [24]. It is not an educational game first. The player finds a phone and has to get access to the phone owner's accounts to learn increasingly more about the owner. It highlights how much can be learnt from a person by simply finding their phone. The game holds a *Very Positive* rating on STEAM.

The game *Casey Joint* is at the intersection of computer and Physical security [25]. In that game the player takes over the role of a hacker who is tasked with supporting a heist crew with breaking into a building. The player needs to unlock doors, complete challenges and hack cameras for the job to succeed.

It can be said that the range of games focusing on computer security is very wide. Many of them utilise unique approaches to tackle content.

CHAPTER 4

Design & Prototype Development

4.1 Game Concept

The game consists of four levels plus a tutorial level with each level focusing on a different security topic.

The levels are:

1. Ransomware
2. Distributed Denial of Service
3. Social Engineering
4. Elevation of Privileges

Every level follows the same game-loop:

1. Player gets introduced to problem statement in terminal
2. Tasks are displayed and have to be executed
3. Mini-game has to be completed (not applicable in tutorial level)

The main screen is identical for every level: A home-screen of a computer with a terminal, list of tasks and two buttons on the bottom. The button *Network Activities* leads to the interactive parts of the game, meaning non-terminal interactions. The button *Terminal* minimises and expands the terminal. The list of tasks adapts to the tasks of the current level. If a player completes a task, the task list scales up and down to steer focus to this part of the game. The main interaction point of the player with the game is the game's

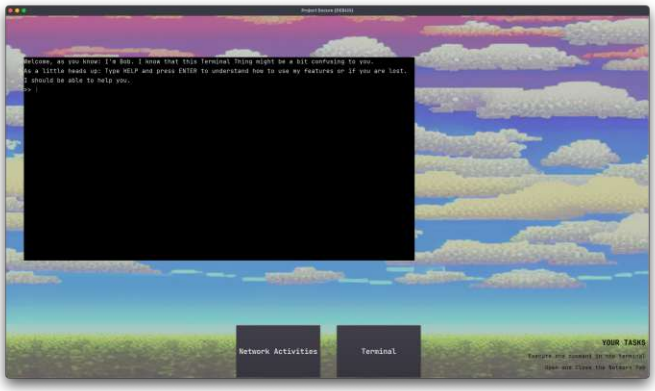

Figure 4.1: Main Game Screen

```
Welcome, as you know: I'm Bob. I know that this Terminal Thing might be a bit confusing to you.
As a little heads up: Type HELP and press ENTER to understand how to use my features or if you are lost.
I should be able to help you.
>> LIST TERMS
I can give you more information on the following terms:
... RANSOMWARE
... FIREWALL
... HONEYPOT
>> EXPLAIN RANSOMWARE
A type of malicious software that tries to gain access to a computer, fetch sensitive data and block
access.
>>
```

Figure 4.2: Tutorial Commands Example 1

terminal. Just like a real terminal, the player can input commands to complete tasks or get more information on topics, see examples 4.2 and 4.3. For more information on supported *Terms*, see 4.2. For more information on all supported commands, see 4.1. These terms are used to explain concepts to the player and introduce them to new topics. The number of supported commands and terms increase with every level. The terminal is also used to convey the story of the game to the player via text.

```
... RANSOMWARE
... FIREWALL
... HONEYPOT
>> EXPLAIN RANSOMWARE
A type of malicious software that tries to gain access to a computer, fetch sensitive data and block
access.
>> HELP
I understand the following commands ...
MIN: Minimises me. Not that you'd ever want to do that.
CLEAR: Clears our entire chat history. Please don't do it if you don't want to hurt my feelings :)
GRAB COFFEE: Humans also need a sometimes break (even though I don't understand why - they're so
unproductive). Might have side effects.
BACKSTORY: I give you a backstory to the current chapter. It's ok. Humans are forgetful.
LIST TERMS: Lists all terms understood by me. The list might expand as you progress in your tasks.
EXPLAIN <TERM>: Explains a specific term. I'm all-knowing, you know.
CREATE FIREWALL: Creates a Firewall used for filtering incoming Network Requests. Necessary to execute
once so I know you really want it enabled.
LISTEN REQUESTS: Tells me that you want me to fetch the most recent requests. Necessary to execute once
so I know you really want it enabled.
TOGGLE HARDMODE: For developers who want the extra challenge. If enabled, increases the difficulty of
challenges thrown your way. Can be disabled any time by re-entering the command.
>>
```

Figure 4.3: Tutorial Commands Example 2

Command	Description	Supported Level(s
MIN	Minimises the terminal.	All levels
CLEAR	Clears all text from the terminal	All levels
GRAB COFFEE	The player grabs coffee. Can lead to side-effects.	All levels
BACKSTORY	The player is presented with the level's backstory.	All levels
LIST TERMS	Lists all terms the game can explain to the player.	All levels
EXPLAIN <TERM>	Explains a term to the player.	All levels
CREATE FIREWALL	Creates the initial firewall.	Ransomware
LISTEN REQUESTS	The machine starts listening to all incoming network requests.	Ransomware
ENABLE IDS	Enables the IDS so that the system can notify in case of intrusions.	DDoS
CHECK CAPACITY	Checks the current server capacity.	DDoS
CHECK IDS	Checks whether the IDS has found requests.	Social Engineering
CHECK EXPLOITS	Checks whether the system has found that it has been exploited.	EoP
TOGGLE HARDMODE	Enables/Disables hardmode for mini-games. Easter egg.	All Levels (- Tutorial)

Table 4.1: Terminal Commands

Term	Description	Supported Level(s
FIREWALL	A type of network protection to filter incoming network traffic.	All levels
HONEYPOT	A type of system to deliberately attract attacks. The system looks normal from an attacker's point of view but it is not used. Instead, it watches the actions of the attacker to give an understanding of attack patterns.	All levels
RANSOMWARE	A type of malicious software that tries to gain access to a computer, fetch sensitive data and block access.	All levels
DDOS	A type of system attack where attackers send a high number of requests to a single server with the goal of overwhelming it and forcing it to go offline.	DDoS Social Engineering EoP
IDS	Intrusion Detection System (IDS) It can be used to detect and alert on suspicious or malicious traffic on the network.	All levels
IPS	Intrusion Prevention System (IPS) It can automatically block malicious traffic in real-time.	All levels
SOCIAL ENGINEERING	A type of system attack to either manipulate people into performing actions or accessing confidential information through other people's accounts. Access can be gained in many ways, i.e. through manipulating people into giving the attacker their credentials.	Social Engineering EoP
EOP	Elevation of Privileges (EoP) A type of attack where attackers gain access to an internal account and elevate the account's privileges to access information that would otherwise not be accessible to the account.	EoP

Table 4.2: Supported Terms

4.2 Game Design Process

4.2.1 Initial Idea

The initial game design idea was based on the author's interest for text based games. The concept of writing text to advance a storyline has always fascinated and captivated them. The player gets to choose their own path and explore the game on their own terms. The author thought that combining writing text with educational content could yield very beneficial results for the player.

The initial idea consisted of the player taking over the roll of a developer. Their workstation would feature a desktop view - similar to the final result - where they can interact with the terminal to progress the storyline. Notably, they would defend the system from external attackers. The player could also leave their workstation and move around freely in an office space. The office would feature several different areas where the player could interact with different Non-Playable Character (NPC)s to advance specific storylines.

The game would focus on topics covered in the course *Denkweisen der Informatik*. The initial idea did not yet cover which exact topics would be featured in the game and what the exact game loop would look like.

The potential topics were:

1. Stackelberg Security Games
2. Physical Attacks
3. Distributed Denial-of-Service Attack (DDoS)
4. Social Engineering
5. SQL Injection
6. Zero Day Exploit
7. Ransomware

4.2.2 Initial Discussions & Feedback

Supervisor Feedback

The idea was presented to the supervisor with the initial reception being very positive. However, it was agreed upon to only focus on the workstation aspect as the office aspect would not add much in terms of educational content this concept was removed.

Expert Feedback

The idea was presented to Professor Edgar Weippl [4]. The main aim of this interview was to understand his approach to gamified education and to receive feedback on the idea. He mentioned to clearly define the learning outcomes of the game: What should the game convey to the player? He also suggested to focus on specific topics, i.e. Elevation of Privileges (EoP). He suggested that the game could feature different scenarios were the player would have to drag and drop actions in the right order. The author suggested that the game could feature multiple levels with each level focusing on a specific topic. This would become an integral part of the game's design.

Focus Group Discussion

The amended idea featuring different levels was then presented in a focus group session with the *Security and Privacy Group* at TU Vienna [5]. The general feedback was very positive. Several ideas for game content were brainstormed. Topics featured in *OWASP Top Ten* were discussed [26], notably, how to gamify permission systems and password security. It was mentioned that the game could feature a view imitating an admin view where the player could change users' permissions. The general idea of permission systems would later be used in the game.

The designs of other educational security games were also discussed. At the end of the session it was concluded that the presented idea seems promising. The main takeaways of the session were the discussions on gamification of security topics.

Main Takeways

The author had many takeaways from discussing the author's idea and education games in general. The general feedback for the game idea was very positive. However, more focus should be put on understanding what the game should convey to the player. The game should be split up in levels with each level focusing on one topic. The topics should be well selected and relevant. For example, they could be selected from *OWASP Top Ten* [26]. The author opted for topics covered in the course *Denkweisen der Informatik* to keep it relevant for the players whilst prioritising those which are relevant according to *OWASP Top Ten* [26].

4.2.3 Iterative Design Decisions

First Design Iteration

The topic for the first level was selected after reviewing the relevant topics in the course *Denkweisen der Informatik*: Ransomware. The first implementation of the prototype consisted of a bare implementation of the terminal. A common theme and main enemy were brainstormed: The player would defend the company from cups which is a nod to the stereotype that developers drink a lot of coffee. It was quickly realised that a tutorial is necessary to introduce the player to the game and present the player-motivation.

Focus was first put on doing research on the topic of Ransomware. Ransomware executes malicious requests on a computer or a network.

The game would feature a view where the player would be able to see and interact with every single request within the network. This would be called the *main hub*. They would be able to see every connected computer and user with their actions at any given moment. The *main hub* would feature different visualisations, each focusing on a different aspect, see 4.4 and 4.5.

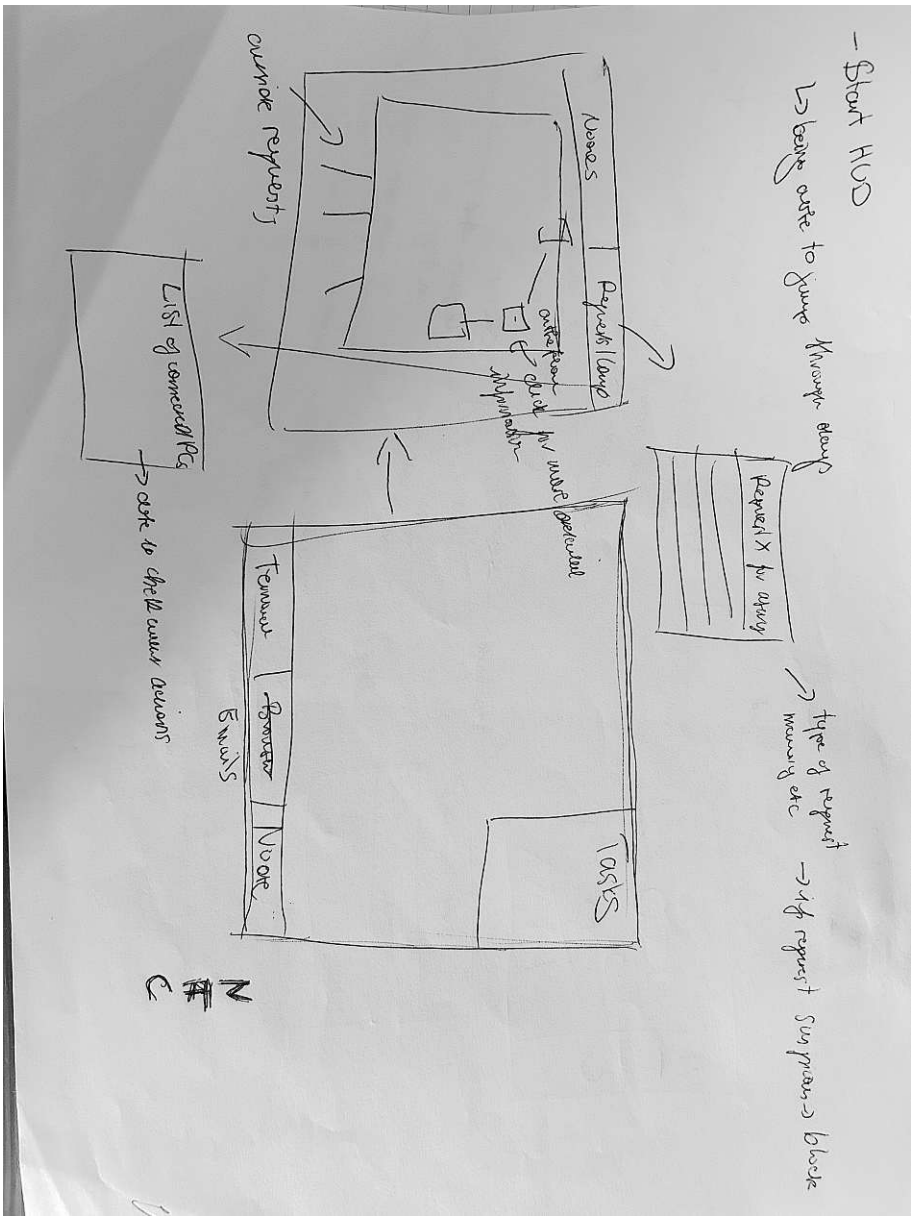

Figure 4.4: Initial HUD Idea

The main visualisation would visualise all connected entities as a node network. Incoming and outgoing requests would be visualised as lines. The player could then hover over the different computer nodes to learn more about the user's activity. Malicious requests would be highlighted as red. The player would then easily see which requests were malicious. The second visualisation would display all connected computers in form of a list. The third would display all requests in form of a list showing different attributes, i.e. origin and bandwidth. The player would also be informed of malicious requests in forms of a popup in the main view.

The *main hub* would also feature a *Manage* section where the player would be able to setup the honeypot to create an isolated environment to attack malicious requests, see 4.6.

The approach of Ransomware would go hand-in-hand with the topics DDoS and Social Engineering. As discussed by Wang et al. (2020) the line separating different kinds of computer security attacks is getting increasingly blurry [27], [28]. The network view would also feature a system health tab to monitor server capacity, which links to DDoS. Player tasks would be handled in the form of incoming emails. This could then be reused for the section focusing on Social Engineering. An attacker would send employees a malicious email and the player would have to block it.

The game would also focus on the topic EoP since it goes together with Social Engineering. The topic was found by reviewing the potential topics mentioned in 4.2.1 and by reviewing literature. In the game the player would have to review requests from users and check for user actions and permissions that do not go together.

However, upon presenting this to the supervisor, the author realised that this concept was convoluted. The concept would not allow for a clear separation of levels and themes. Also, the concept might proof overwhelming for players. Additionally, these ideas would lead to the game barely focusing on terminal interaction when these interactions should be an integral part of the game. Therefore, most ideas of the first design interaction were discarded. The idea of the incoming requests was later reused in the level Ransomware. General concepts - e.g. which topics work well together - were also reused. The handling of incoming tasks was simplified to a simple box showing current tasks. The idea of handling tasks via emails would introduce too much overhead for the player so that approach was discarded as well.

Second Design Iteration

The second design iteration was supported by a working prototype of the terminal, a tutorial and the first level focusing on Ransomware. Therefore it already featured a working implementation of the main game-loop.

In the tutorial the terminal only supported the command *MIN* to minimise the terminal. This was implemented so that users can complete the first task, which is to execute a command in the terminal. The main storyline was also introduced with *cups* being introduced as the enemy. The enemy was not yet visualised in form of assets.

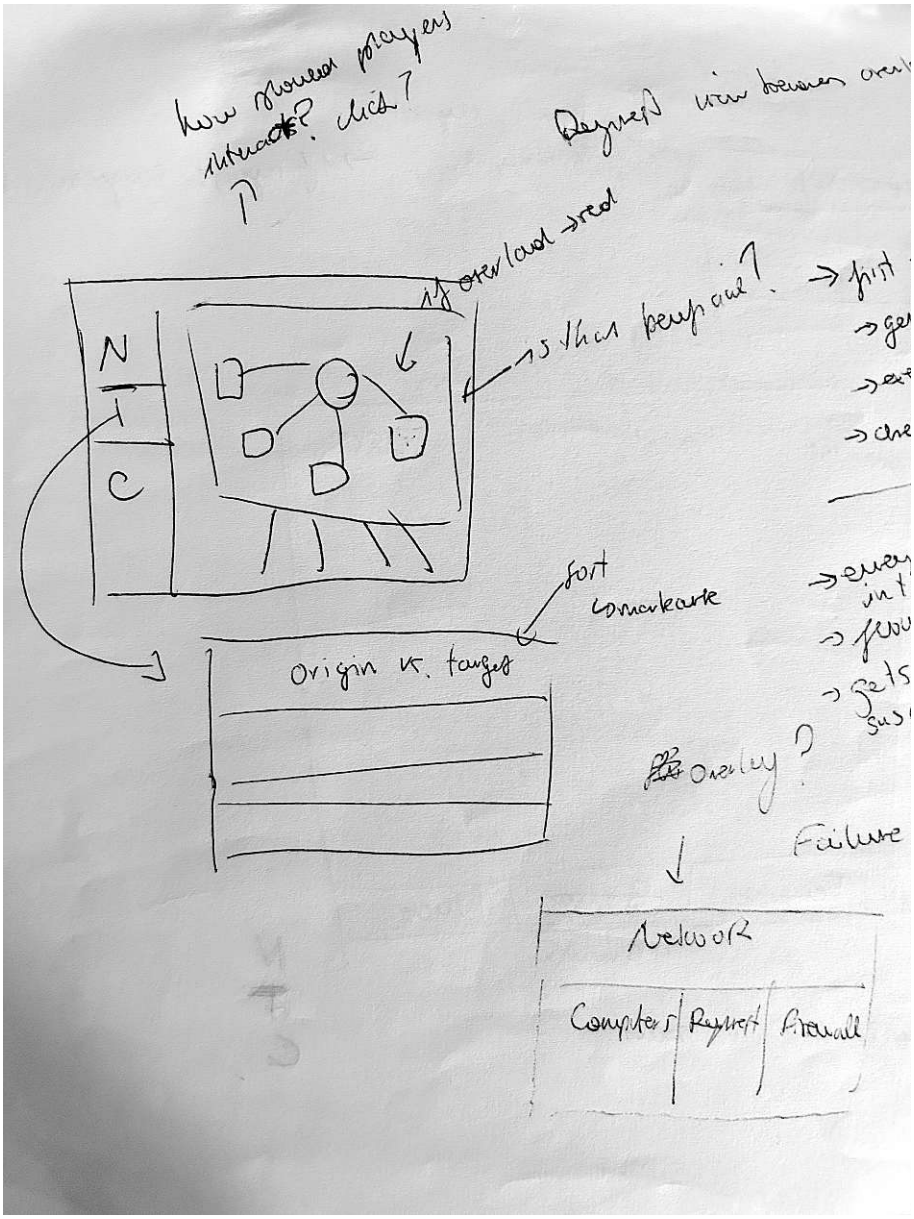

Figure 4.5: Initial Node Network Idea

The first level featured the mini-game as it is present in the final version of the game in a basic form. The requests did not appear from the top but were static. Filters also were not implemented yet. Key tasks were also added which were required to complete the level. To achieve this, the terminal supported commands *CREATE FIREWALL* and *CREATE HONEYPOT* to immediately complete these tasks. Interaction with pop-up windows - which are used in the final prototype - to complete the level was not yet implemented. This will later be covered in 4.4.2.

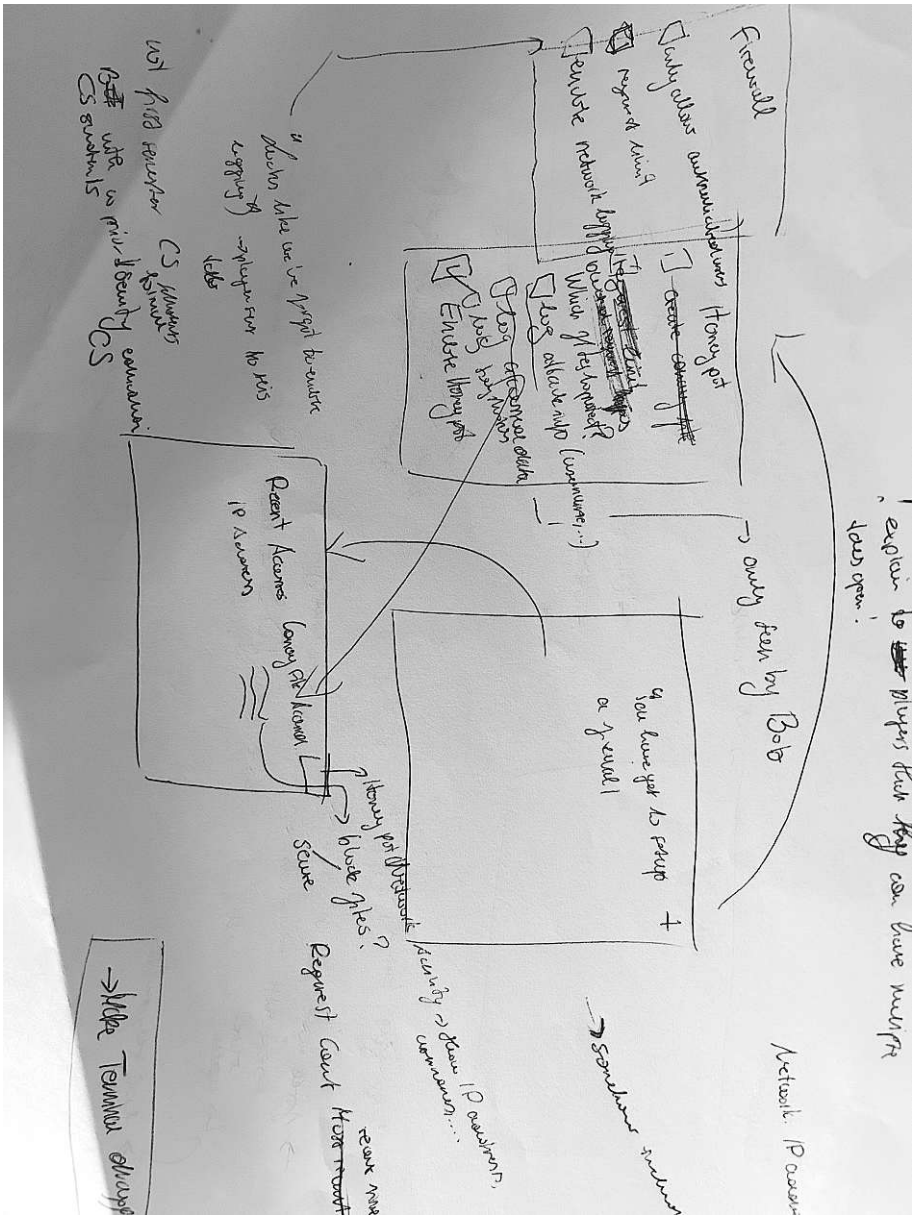

Figure 4.6: Initial Firewall Idea

The main open questions of this prototype were:

1. How to connect the different themes/levels?
2. How to introduce these concepts to players in an efficient manner?
3. How much time should the game take?

Feedback from the supervisor to this prototype was very positive. The ideas from the discussion about the other levels mostly consisted of introducing new windows to complete tasks and levels. It was agreed upon that game duration of the prototype is secondary.

For the level Social Engineering the initial level trigger idea was an incoming phishing mail that needed to be blocked but another employee had already interacted with the mail, introducing the attacker to the system. For EoP the idea consisted of opening network requests and marking suspicious requests. The main focus would be on different types of requests, e.g. where the attacker is coming from. As already discussed in 4.2.3, focus on terminal interaction would be lost if the player had to juggle several different tabs.

Based on this, the design had to be chosen in a way that supports the terminal. Upon discussing this with the supervisor, it was concluded that the game should feature several mini-games which highlight the theme of the respective level. This way the player has to interact with the terminal in every level and separation of levels is clear.

The levels would not necessarily have to convey dense information but convey the basics of the respective topic. The mini-game as it was implemented for the level Ransomware could be left as is. More complex systems like the mail and node systems presented in 4.2.3 should be discarded.

The initial idea for the level DDoS was a *Space Invader* inspired mini-game where the player had to defend the system from attackers. This idea ended up being in the final version of the prototype. Ideas for the mini-games part of levels Social Engineering and EoP had yet to be developed. These were part of the third and final design iteration.

Third Design Iteration

The third design iteration of the game is almost identical to the final version of the prototype apart from bug and text fixes. The game was segmented into four levels which were mentioned in the beginning of 4.1. Each level had its separate tasks and mini-game. Levels Ransomware and DDoS had already developed level ideas prior to this iteration. The games assets were created. In particular, the enemy was visualised in form of different assets. In addition different transition animations were created which featured the enemy *cup* in an action that should introduce the player to the subject. For example, the transition to Social Engineering would feature the *cup* with a disguise, see 4.7. The animation would be displayed right after the level has started on the right side of the terminal.

In the level Social Engineering the mini-game would feature a memory game where players had to match user to network action. At the end of the game a pair would remain where user and action would not fit. This was developed for players to understand expected user actions versus suspicious user actions. To finish the game, the level EoP would feature a jump and run section where the player had to catch the enemy in order to end the game.

Figure 4.7: Social Engineering Transition Frame

An important step of this design iteration was the implementation of an overlapping story arc to the game. The storylines of the levels would build on top of one another. As an example, the chapter Ransomware leads to the chapter DDoS. This is explained in the game with the attacker breaching the network and seeing parts of the system. The attacker's next goal is to take the system offline. The player needs to enable IDS and IPS to defend the system. The chapter DDoS then leads to the chapter Social Engineering. The game explains to the player that some employee must have shared their account information with the intruder. The attacker is exposed via the system's IDS. The player now needs to find the exposed account. The segue-way to the last level EoP is the player having collected enough information about the attacker. The player can finally pin the attacker down and end the attacks.

As aforementioned, the storyline is supported with relevant tasks and mini-games. Some tasks needed to be completed in the terminal, some in overlays. Not every mini-game conveys dense information to the player. This was a deliberate choice to keep the game more entertaining. It already featured dense information in the context of the terminal.

Feedback from the supervisor to the final design iteration was very positive. This marked the end of the design phase.

4.3 Technical Implementation

The game was implemented using the *Godot* engine version 3.5.5 [6]. The enemy assets were created using Aseprite [7]. The background asset for the mini-game featured in EoP is a free asset taken from *itch.io* [29]. The other assets featured in this mini-game were taken from the default asset package in *Godot*. The background image for the desktop view was generated using the application *DiffusionBee* and entering the prompt: *wallpaper pixelart sky with several clouds* [30]. The chosen font is the free and open source font *JetBrainsMono* [31]. The game's source code can be viewed on Github [32].

4.4 Prototype Showcase

The player first starts the game on the main screen and needs to interact with the *Start Game* button to start the game, see 4.8. After interacting with the button, the

Figure 4.8: Intro Screen

player is prompted with an introductory text explaining the context of the game, see 4.9. The main aim of this view is to convey the *motivation* behind the story to the player and describe the main interaction points of the game. Certain terminal commands are mentioned as well as key sections as the location of the task list. Dismissing this view leads to the tutorial section of the game.

4.4.1 Level: Tutorial

Learning Goals

The player has to complete the following tasks in this level:

1. Execute one command in the terminal
2. Open and Close the Network Tab
3. Grab some coffee

The main aim of these tasks is to get the player accustomed to the main game interactions. These consist of interacting with the terminal and understanding what the button *Network Activities* is used for.

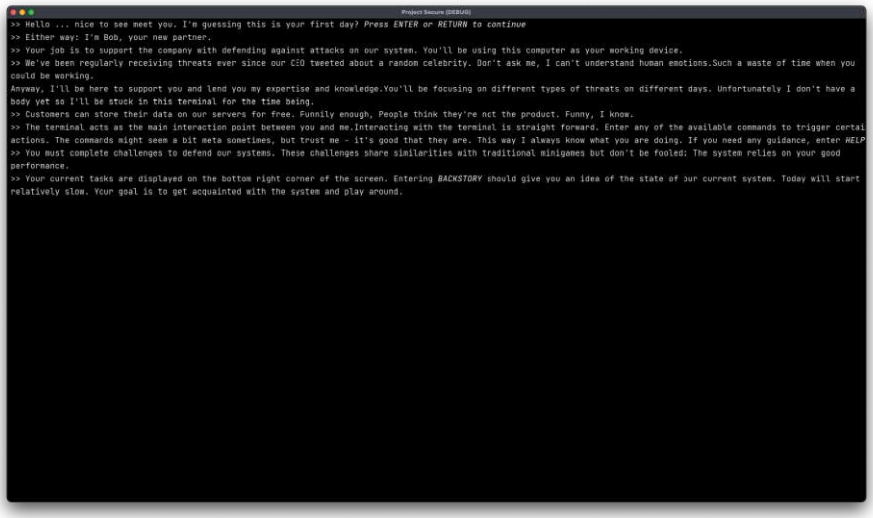

Figure 4.9: Introduction to Game

Level Content

The player is presented with the main view of the game for the first time, as can be seen in 4.1. The player then needs to progress in the tasks provided. For the first task, the player simply needs to execute any command. For the second task, the player needs to open and close the *Network Activities* button. When opening the *Network Activities* button the player is prompted with the message *Network information will be displayed here.*. This indicates that the player should keep this button in mind for future tasks.

Once these tasks have been completed the player is presented an ominous message appears in the terminal. The player must execute the command *GRAB COFFEE*. Whilst the player is drinking coffee, a player's colleague finds a USB stick. Before the player can interfere, the colleague plugs the USB-stick into their machine, see 4.10. This lays the foundation for the rest of the game's story and marks the end of the tutorial.

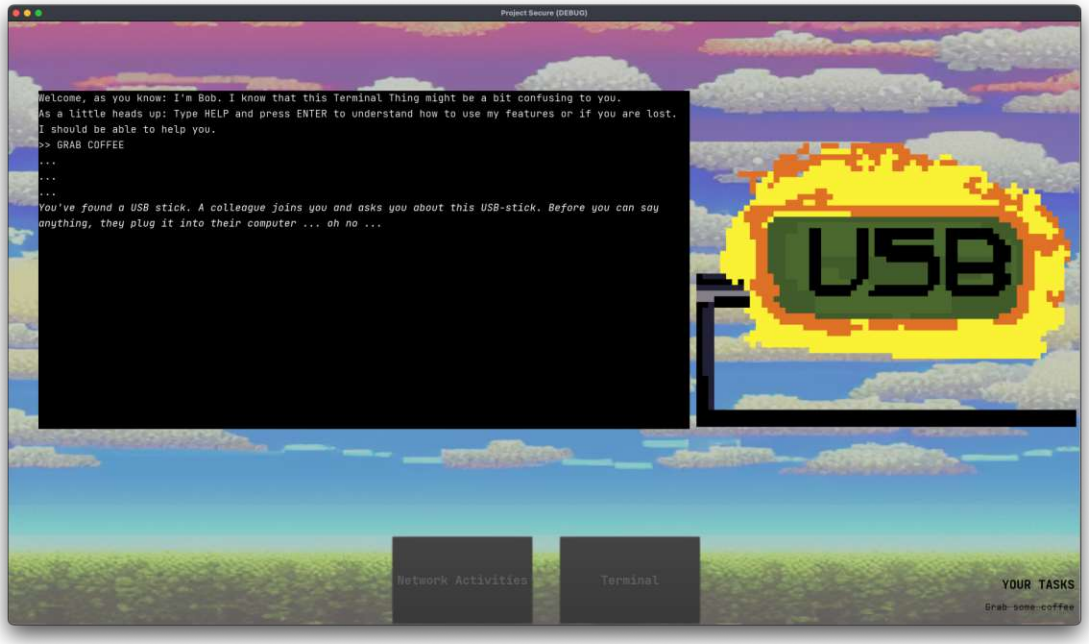

Figure 4.10: End of Tutorial

4.4.2 Level: Ransomware

Learning Goals

The player has to execute the following tasks in this level:

1. Create Firewall
2. Listen to Network Requests
3. Enable Firewall
4. Enable Honeypot
5. Mark Suspicious Internal Requests

The main aim of this level is to introduce the player to the topic of Ransomware. In addition, the player is introduced to the concepts of Firewalls and Honeypots to create a better understanding of the topic. Finally they learn how to separate a malicious network request from a non-malicious one.

Level Content

The level starts with the player being prompted with the first two tasks, they need to create a firewall and listen to network requests. Both of these tasks are done in the terminal. Tasks *CREATE FIREWALL* and *LISTEN REQUESTS* have to be executed. After successful completion of these task, two follow up tasks appear. The player needs to enable both the firewall and the honeypot. To achieve this, the player has to open the *Network Activities* button, followed by the *Firewall* button. The player is presented with a pop-up dialogue with several options to check, see 4.11. For the Honeypot, the player needs to do the same. Notably, they also need to enter a name for the lure file. After the

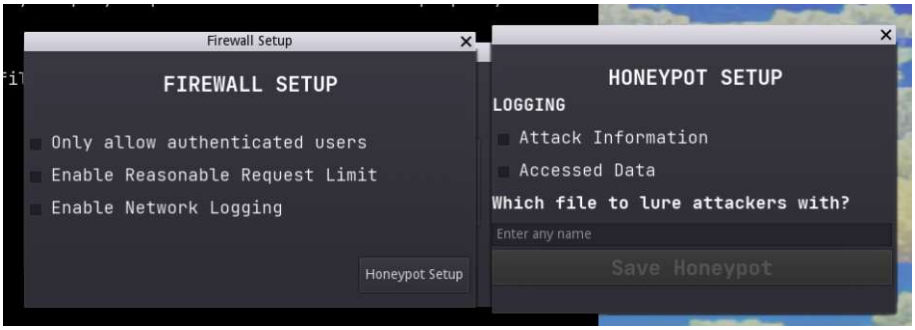

Figure 4.11: Firewall & Honeypot Setup

completion of these tasks, to the player is ready to start the mini-game, see 4.12. This is done by opening *Network Activities* followed by *Activity*.

The aim of the mini-game is to find the malicious request amongst the normal requests. The requests appear at the top of the network request list every 1.8 seconds.

Each request contains five different attributes:

1. Origin - IP Address
2. Requested URL - What is the request trying to access?
3. Role - Who is executing the request?
4. Type - Where is the request coming from?
5. Count - How many times has this request been executed?

The requests are removed once they reach the bottom of the view. The requests have to be selected and moved to the *Blocked Requests* section (right side) with the respective button. Columns can be shown and hidden by interacting with the filter attributes on the left side of the mini-game screen. The player is also able to filter requests by entering text in the text box on the left side. The text is associated with the request texts listed in the column *Requested URL*.

If the player blocks a normal request, they need to move it back to the *Incoming Requests* section. To facilitate the selection process, the player can hide certain columns or filter by text. On the top right a counter is displayed which starts at 100. Every second, the counter deducts 2.5. Once the counter reaches 0, the game is lost and needs to be restarted. The game is won if and only if the right request is blocked. If the mini-game

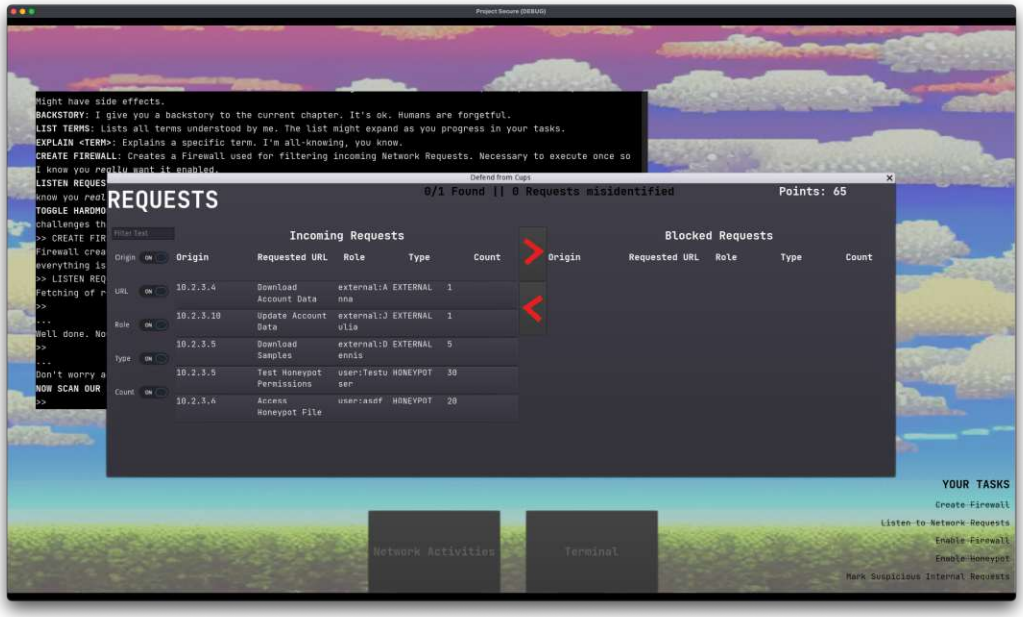

Figure 4.12: Ransomware Mini-Game

has been completed successfully, the player is presented with a win-screen and the message *Interesting things that are going on in the honeypot ....* This indicates to the player that more things are going on in the background. After interacting with the screen, the player is automatically forwarded to the next chapter.

4.4.3 Level: DDoS

Learning Goals

The player has to execute the following tasks in this level:

1. Enable Intrusion Detection System
2. Check the server's capacity
3. Defend against the attackers

The main aim of the level is to introduce the player to the concept of server capacity and tools against attacks on server stability, i.e. IDS and IPS. The tasks are formulated in a manner that the player has to executed command *HELP* at least once.

Level Content

The level starts with the player being introduced to the problem statement: The system is presumably under attack. The player is tasked with enabling a detection system. This is done by executing the command *ENABLE IDS* in the terminal. The server's capacity also needs to be checked so that the player is made aware of the urgency of the problem. Once these tasks have been completed, the player must defend against the attackers. This is done in form of a mini-game.

In the mini-game, the player must shot down cups which are trying to get into the system, see also 4.13. The system starts with 100 health points. A cup spawns every 1.2 seconds. If a cup reaches the system (bottom of the windows), the system loses 5 health points. For each destroyed cup, the player gains 5 points. The player must survive 20 seconds to win the mini-game. After successfully defending against the attack, the player is presented with a win-screen displaying the message *You showed these cups! But wait ....* This indicates that something is still off within the system.

Figure 4.13: DDoS Mini-Game

4.4.4 Level: Social Engineering

Learning Goals

The player has to execute the following tasks in this level:

1. Check if an intruder has breached the system
2. Safe the system!

The main of this level to convey the implications of *Social Engineering* to the player. Meaning, which implications a network breach can have on the integrity of user accounts, i.e. users trusting alleged in-house mails. Additionally, it highlights the importance of understanding expected user actions versus suspicious user actions.

Level Content

The level starts with the player being informed that the internal IDS might have detected an intruder. It hypothesises that employees have shared their account information. Now it's for the player to check this. To achieve this, the player needs to execute *CHECK IDS* in the terminal. This unlocks the level's mini-game.

The mini-game consists of a memory game where the player has to match person (name and occupation) with their network action (which action was registered in the network), see also 4.14. Once every correct pair has been found, the player is left with two cards where person and action do not match. In the case of this game, is indicates an elevation in privileges. The game conveys to the player that this action is associated with the attacker. Also, it tells the player: *We finally know what the attacker is up to.* and with that the player is progresses to the last level.

4.4.5 Level: Elevation of Privileges

Learning Goals

The player has to execute the following tasks in this level:

1. Check for exploits
2. CATCH THE INTRUDER

The main aim for this level is to finalise the story and convey the basics of EoP to the player. Meaning, how they can happen and explaining a common example.

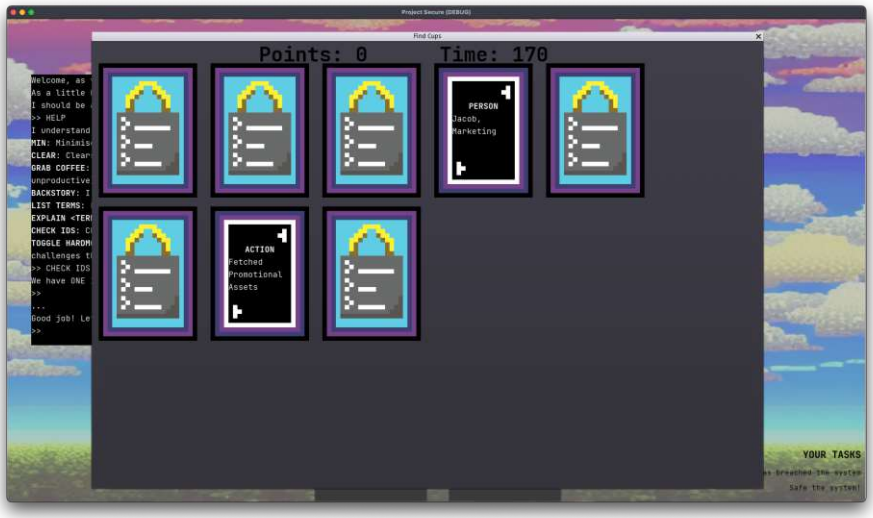

Figure 4.14: Social Engineering Mini-Game

Level Content

The level starts with the terminal telling the player that the systems have collected enough information about the intruder to finally be able to pursue them. The player needs to check for exploits one last time in the terminal with *CHECK EXPLOITS*. This tells the player that the attacker conducted a database exploit and that future actions have to be done by a different team:*Hm ... seems like the attacker elevated their privileges by using a simple database exploit. I need to delegate this to Marvin. My fellow database colleague.*. Executing this unlocks the final mini-game.

The mini-game is a jump-and-run game where the player takes over the roll of the terminal and needs to catch the cup, see also 4.15. The characters start on different platforms, the player starts on the bottom platform. However, once the player has successfully traversed all obstacles, the player can jump to the top platform and catch the cup to end the game. This marks the game ending. At the end of the game, an overlay appears telling the player that they have saved the system, see 4.16. It also shows a destroyed cup. Upon interacting with the overlay, the player is forwarded to the post-game evaluation and sent back to the Main Screen, see 4.8.

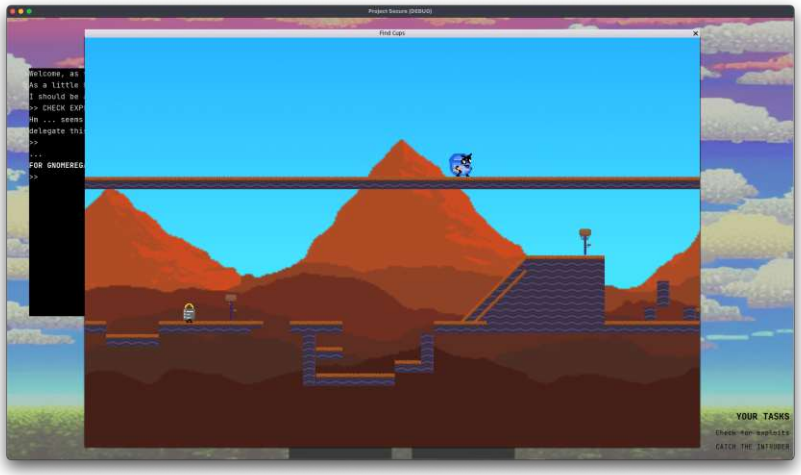

Figure 4.15: EoP Mini-Game

Figure 4.16: Game End Screen

CHAPTER 5

Game Evaluation

5.1 Expert Feedback

The game was presented to the security and Privacy Group at TU Vienna [5] on 13/4/2023. A link to the game was shared a couple of days before the presentation in order for participants to be able to play the game themselves. The presentation started with explaining the motivations behind this topic, followed by a chapter-by-chapter walk-through of the game.

The presentation was followed by a 30-min discussion on several topics. One piece of feedback was that the first level gamified the blocking of a request very well. This sparked a discussion about why a game like the one presented would be preferred over a game similar to a Capture the flag (CTF) competition where players have to defend and attack other groups participating in the same competition. CTF can be seen as a safe environment to learn hacking. It was argued that CTF takes a significant time to setup and player interest might be lost, especially when they are new to the field.

It was also argued that some approaches presented would be too gamified and that they would most likely be bored by it. This sparked an even bigger discussion. Another member of the audience argued that someone with a good understanding of the computer security is not the target audience of this game. After a lot of back and forth, a conclusion was reached that the game design is effective enough for the target audience.

Another point that was mentioned was how the author ensured that players focused on the correct things when playing the game. The author replied that they could not ensure that. However, the pre- and post-game questionnaires feature content specific questions which help understand which knowledge the players retained. An audience member made the counter argument that asking the same questions pre- and post-play could potentially lead to inflated player focus. Whilst the author agreed on this, the audience member

agreed that this is a tricky issue and that the presented approach should be sufficient for the present study.

One of the last points of discussion was to review how other educational computer security games gamfied different concepts. In particular, the game *Uplink* was mentioned and that it was one of the first successful educational computer security games [23].

To summarise, feedback based on the presentation was very positive. Several interesting discussions sparked from the presentation. The important question of how to handle player-focus was brought up as well as how to guide players. Interestingly, it took the audience several minutes to agree on them not being the target audience of the game. This can possibly be explained with experts being biased when it comes to newcomers to the field. In general, audience members received the game positively. Their feedback was useful when evaluating the feedback from players both in the questionnaire and playtesting sessions.

5.2 Player Questionnaires

Two questionnaires were created, one pre- and one post-play. For a general overview of the player questionnaire approach, please see 2.5.2. The main aim of the split questionnaires was to spot a difference in player-knowledge and player interest in the topic of computer security pre- and post-gameplay.

5.2.1 Questionnaire Structure

Pre-Game Questionnaire

The pre-game questionnaire consisted of ten questions, four of which were open-ended questions and six of which were single-choice questions. The questionnaire was split into two sections. The first section contained two general questions. The questions were focused on understanding the player's connection to computer security. The questions were:

- How would you rate your knowledge in the field of computer security? (Q1)
- Are you planning on taking any non-mandatory security courses?

Q1 utilised a five-level Likert-type scale: 1=*None*, 2=*Minimal*, 3=*Average*, 4=*Considerable* and 5=*Expert*. The second question offered three response possibilities: *Yes*, *No* and *I don't know*.

The second section consisted of eight *knowledge checkup* questions, four of which were single-choice, four of which were open-ended questions. The questions revolved around topics covered in the game with two questions revolving around one chapter each. The first question for each chapter was a single-choice question revolving around how a player

to ask the player would approach a specific scenario, followed by an open-ended question where they could describe their strategy in mentioned scenario. If an open-ended question was left empty, the author interpreted it as the player not knowing the answer. The question were:

- How confident are you in defending your network against an attacker? (Q2)
- What would you do to defend your network against an attacker?
- How confident are you in defending your network if an attacker has breached it? (Q3)
- What would you do if an attacker has breached your system?
- How confident are you in educating your hypothetical employees about Social Engineering? (Q4)
- How would you educate your hypothetical employees about Social Engineering?
- How confident are you in handling a user that has elevated their privileges? (Q5)
- What would you do if you notice that a user has elevated their privileges?

The single-choice questions followed a four-level Likert-type scale:1=*I would not know what to do at all.*, 2=*I would have a slight idea of what I need to do.*, 3=*I would have a concrete battle plan but wouldn't know how to handle certain scenarios.* and 4=*I would be very confident in my actions and have a backup plan for most scenarios.*.

Post-Game Questionnaire

The post-game questionnaire consisted of 19 questions, eight of which were open-ended questions and eleven of which were single-choice questions. It was split into three sections: general section, content section and game related section. The questions in the general section were focused on understanding whether playing the game has changed their interested in the field of computer security and how much they took away from playing the game. The questions were:

- Has playing the game changed your interest in the field of computer security?
- Did a particular chapter peak your interest in a certain topic? If so which one and why?
- How much did you take away from playing the game? (Q6)
- How would you rate your knowledge in the field of computer security? (pre-gameplay)

- Are you planning on taking any non-mandatory security courses?

The questions in the content section were identical to the ones asked in the *knowledge checkup* section of the pre-game questionnaire. This was done to detect any changes in knowledge. Q6 used a four-level Likert-type scale: 1=*Little to none - I didn't take away anything*, 2=*A bit - I took away some basic concepts*, 3=*Moderate - I took away some details*, and 4=*A lot - I took away details from every chapter.*.

The questions in the game related section were focused on gathering feedback on the game. Meaning, which parts of the game were and were not enjoyable, how well the content was presented and general game feedback in form of an open-ended question. The questionnaire closed with a single-choice question asking the player whether they took something away from playing the game. The exact questions were:

- What was your favourite level?
- Which part didn't you enjoy and why?
- What was the most enjoyable aspect of the game?
- How well was the content presented? Was the game-play easy to follow? (Q7)
- General Game Feedback
- Do you have the feeling you took something away from playing the game?

In particular, Q7 used a four-level Likert-type scale: 1=*It wasn't well presented, I had trouble following the levels.*, 2=*It was decently presented. However, sometimes I had issues following gameplay.*, 3=*It was presented in a good manner, I rarely had trouble following gameplay.*, and 4=*It was very well presented and easy to follow.*.

5.2.2 Questionnaire Results

The pre-game questionnaire received 34 responses and the post-game questionnaire received 12 responses. The discrepancy between pre- and post-game responses can be explained with the procedure taking time. The questionnaires were open between 28/03/2023 and 15/05/2023. This section will first discuss the different questionnaire section results separately and will then summarise all results. For the results of the Likert-type questions Q1-Q5, see 5.1. For the results of Q6 and Q7, see 5.2. It should be mentioned that the results are not statistically significant given the limited number of replies.

General Results

The initial question of the pre-game evaluation was about player-security knowledge (Q1). This question is crucial in understanding who played the game. As previously mentioned, players needed to self-assess their level based on five given options. Only one person answered that they have no experience with security. Most people - being 20 - answered that they have either minimal or average knowledge of security. The remaining 13 people answered that they either have considerable or expert knowledge on computer security. The same post-game answer yielded different results. Six people answered that they have minimal knowledge whilst the remaining six responses were answered with *Average* or above. Only one person answered with *Considerable* and *Expert* respectively.

Based on the results of this question, it can be said that most players had average knowledge of the field of computer security. Players who completed the game were more likely to have less knowledge of security. This is understandable given the target audience of the game. The game was not designed for people with considerable computer security knowledge.

To understand the player's academic computer security motivation, the questionnaire asked whether the player is planning on taking any non-mandatory security courses. Here, the distribution of answers for pre- and post-game evaluation were nearly identical. Replies *Yes* and *No* were perfectly balanced for the pre-game questionnaire and split 55/45 in favour of *No* for the post-game questionnaire. Based on the balance of the responses, it can be said that both people with academic computer security motivation and people with none have contributed to the questionnaire responses. It should be noted that self-assessed player computer security knowledge did not seem to have an effect on this answer.

The post-game questionnaire contained three additional questions which were not part of the pre-game evaluation. The first of these questions inquired about change in player interest. Five people answered that their interest for the field increased, whereas the other seven answered that their interest remained the same. This would imply that the game increased the interest of 41% of players. This number is acceptable when comparing these answers with the results of the self-assessment. As mentioned when discussing the first question, six people answered with *Minimal* when assessing their knowledge. Four of these six people saw their interest level increase after playing the game. The interest increased for one person who has average knowledge. This proves that the game is effective to an extent in making the topic computer security attractive to newcomers.

The last question of the general section of the post-game evaluation asked about any topics that peaked the player's interest. Honeypot was mentioned four times. People in particular enjoyed identifying an attacker. Social Engineering was mentioned twice. Both IDS and EoP were mentioned once. To be more precise, the player who answered EoP had not heard about that term before. Four responses were left empty. The results show that the topic Honeypot which is covered first level saw the best reception - with players enjoying the interactivity of the level.

To summarise the game was primarily played by people with average or minimal knowledge in the field of computer security. Players were split on taking a non-mandatory computer security course. The less computer security knowledge the player had, the more likely it was that playing the game increased their interest in the field. The topic that was best perceived was Honeypot. This level this topic is a part of has the most interactivity of any level.

Content Results

The first question of the content part was about the level Ransomware: *How confident are you in defending your network against an attacker?* (Q2). Participants could choose from four different answer possibilities. In the pre-game questionnaire, 50% of people answered that they would have a slight idea of how to handle the scenario. Nine people answered that they would have no idea at all. Interestingly, in the post-game questionnaire these two options saw the exact number of replies. Both times 5 people - 41% of players - answered with either reply. This is most likely related to people with less computer security experience being more likely to have completed the questionnaire.

The single-choice question was followed by an open-ended question covering the same topic: *What would you do to defend your network against an attacker?*. The pre-game questionnaire question had 21 complete answers and 13 were left empty. The term "Firewall" was mentioned in ten of these replies. "Secure passwords" was mentioned four times. Two responses mentioned contacting an expert. All of these approaches are valid. The quality of responses was very high in general. This is related to eight out of 13 responses being left empty by participants who answered that they would not know what to do at all in the previous question. In the post-game questionnaire, this question saw seven complete answers and five empty ones. All of the replies mentioned firewall in their answers. This makes sense given the corresponding level in the game - Ransomware - covering this extensively. This leads the author to believe that the game had an effect on player retention. However, no noticeable difference in answer quality could be detected.

The second single-choice question was DDoS themed: *How confident are you in defending your network if an attacker has breached it?* (Q3). In the pre-game questionnaire, 50% of players answered that the would not know what to do with only 17% having a concrete idea of how to approach this. In the post-game questionnaire only 41% of players would not know what to do whilst 41% would have a slight idea of how this should be tackled. This indicates that players have learnt something whilst playing the game.

The open-ended question was as follows: *What would you do if an attacker has breached your system?*. The pre-game questionnaire saw 17 complete answers and 17 empty ones. Eleven of these empty responses came from participants who did not know what to do in that scenario. Similar to the previous open-ended question, the quality of responses was very good. "Taking down the system" was mentioned seven times, whilst "system analysis" was mentioned four times, both of which are valid responses. The post-game evaluation saw seven complete answers and five empty ones. Three of the responses

mentioned checking what was accessed and two of the responses mentioned using IDS whilst one mentioned using a Honeypot. In general, the questions were of lesser quality than the ones from the pre-game questionnaire despite players being more confident in their knowledge. This could be explained by self-assessment being positively biased or players losing motivation.

The third single-choice question focused on the level Social Engineering: *How confident are you in educating your hypothetical employees about Social Engineering?* (Q4). In the pre-game questionnaire 53% of participants answered that they would have a slight idea of what to do in that scenario. Only 17% of participants would not know what to do at all. Very similar to the pre-game questionnaire, in the post-game questionnaire only 16% of participants would not know what to do at all. The ratios of responses were very similar. This could be explained by Social Engineering being an increasingly important topic in every day life.

The corresponding open-ended question was as follows: *How would you educate your hypothetical employees about Social Engineering?*. The pre-game questionnaire saw 18 complete answers and 16 empty ones. As opposed to the previous open-ended questions, only six empty answers were from people that answered that they would not know what to do. Eleven of the responses contained at least a reference to employee training or knowledge transfer regarding password security. “Phishing“ was mentioned twice. It appeared that people had a good understanding of the topic already. The post-game questionnaire question had twelve responses. However, two people wrote as answers that they would not know how to handle the scenario. Three people mentioned “employee training“ and two people mentioned that a gamified approach should be used. One person mentioned not plugging in random USB-sticks, which is a nod to the transition between Tutorial and the first level. Again, the quality of responses was very similar between the pre- and post-game questionnaires, the difference being that two participants saw the value of utilising games for teaching this issue to their hypothetical employees.

The fourth and last single-choice question focused on the level EoP: *How confident are you in handling a user that has elevated their privileges?* (Q5). In the pre-game questionnaire 47% of participants would not know how to handle the situation and only 32% would have a slight idea of what to do. In the post-game questionnaire 41% would not know how to handle the situation. 33% would have a slight idea of what to do. Results for the pre- and post-game responses are very similar with the post-game responses having a slight edge.

The last open-ended question of this section was as follows: *What would you do if you notice that a user has elevated their privileges?*. The pre-game questionnaire question had eleven complete answers and 23 empty ones. 16 of the empty responses were from participants who would not know how to handle the situation. Meaning, as for most other questions, the complete answers were from people who knew at least something about the topic. The high number of empty responses is logical given the percentage of participants not knowing how to handle the situation. Six participants answered that they would block the user. Three people suggested that this would indicate a system

breach and that they would analyse the entire system for any other breaches. Both of these two approaches are reasonable. The post-game questionnaire question had only five complete answers with seven being left empty. Five of the seven empty responses were from participants who would not have an idea of how to tackle the situation. Two of the responses mentioned blocking the user. One user mentioned that finding them would be the highest priority. The answers were not wrong per se but the author noticed a decreased quality compared to the pre-game questionnaire answers. This would indicate that the level EoP would need extensions to convey information better.

To summarise, post-game participants were more likely to know what to do in a specific scenario compared to pre-game. Participants were more confident in their knowledge post-game than pre-game. The quality of responses was very high for both pre- and post-game questionnaire answers with the exception of the post-game answers for the EoP themed question. However, as post-game questionnaire answers were more likely to be coming from participants with less computer security knowledge, it can be said that playing the game had an overall positive impact on player knowledge. In the post game answers references to game content were frequent. This leads the author to believe that the game was successful in conveying basic information on the topics covered in the questions. Only the chapter EoP seemed to not successfully convey its main points to the player.

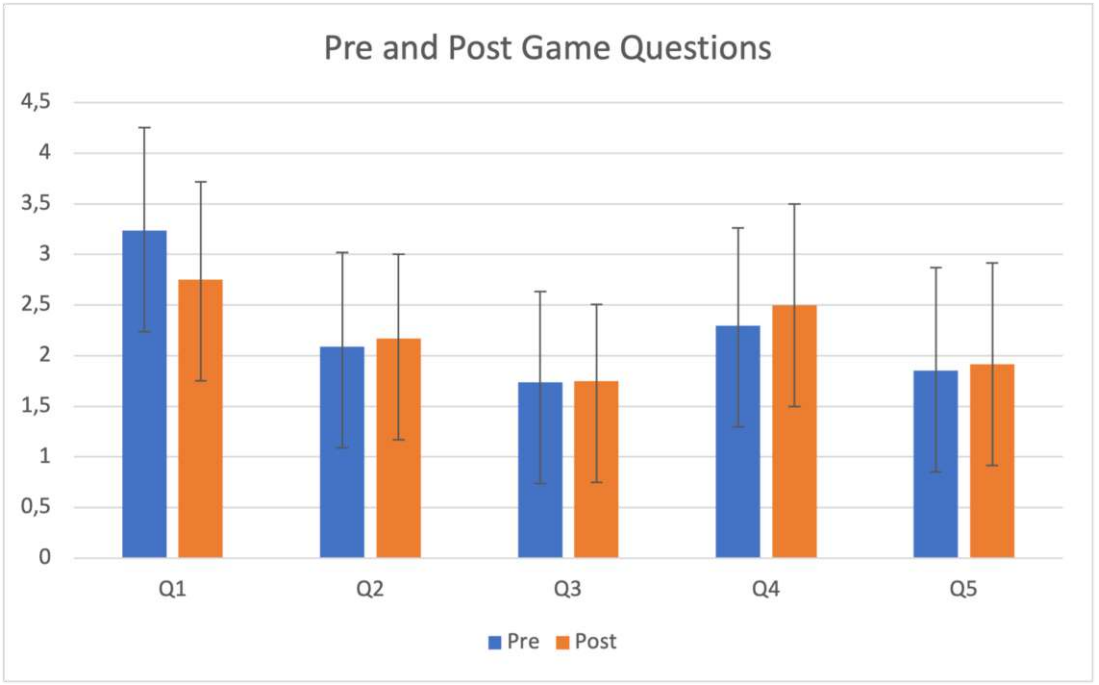

Figure 5.1: Pre- and Post Game Comparison

Game-Feedback Results

The game-feedback questions were part of the post-game questionnaire only. The first question revolved around the player's favourite level. Six people answered DDoS, three people answered Social Engineering, one person answered Ransomware and one person answered EoP. One response was left empty. Interestingly, based on the results of the question covering the favourite topic, the favourite level was assumed to be Ransomware as Honeypot is part of the level Ransomware. However, this was not the case.

When comparing the answers for the question covering which parts the game could improve upon, one bugs was mentioned which resulted in a worse play-experience. The bug was about the game not transitioning between levels Social Engineering and EoP. A card in the memory game would disappear when interacting with it twice leading the mini-game to not be able to be completed. It was also mentioned that the memory game was very slow and that some parts of the game - notably DDoS and EoP - contained very little information. The feedback for EoP makes sense when reviewing the answers for the corresponding content question in the previous section: Not enough information reached the player. One person was unsure what to do in the beginning. The feedback indicates that the game's information should be presented in a better manner. Every level contains additional information in the terminal. However, this is not highlighted to the player. Instead, they have to find the information themselves. This approach did not seem to work.

Reviewing the answers for the most enjoyable aspects of the game, two people mentioned the terminal. One person mentioned the mini-game in the level Ransomware, another mentioned the mini-game in DDoS and another mentioned the mini-game featured in Social Engineering.

The next question asked if the player took something away from playing the game (Q6). Ten people answered with *Yes*, two people answered with *No*. The one person answering with *No* had answered *Minimal* in the self-assessment, whereas the second person to have answered with *No* had answered *Expert* in the self-assessment. The results show that the the target audience of the game - computer security beginners - takes away something from the game. This indicates that the game reached its main goal.

Regarding how well the game was the game was presented (Q7), no person thought that the game was not well presented. Five people answered that the game is decently presented but that they sometimes had trouble following game-play. Five people answered that it was presented in a good manner and that they only sometimes had trouble following game-play. Two people never had issues following game-play. Based on the results of this, the game-interaction should be improved upon. In particular, information about changes in terminal interactions should be better presented to the player.

The author received eight answers for general game feedback. One person noticed that the right side of the screen is empty. They suggested that information about the level could be displayed there. Another person mentioned that some mini-game did not convey any security information. It was also mentioned that the game was sometimes difficult to

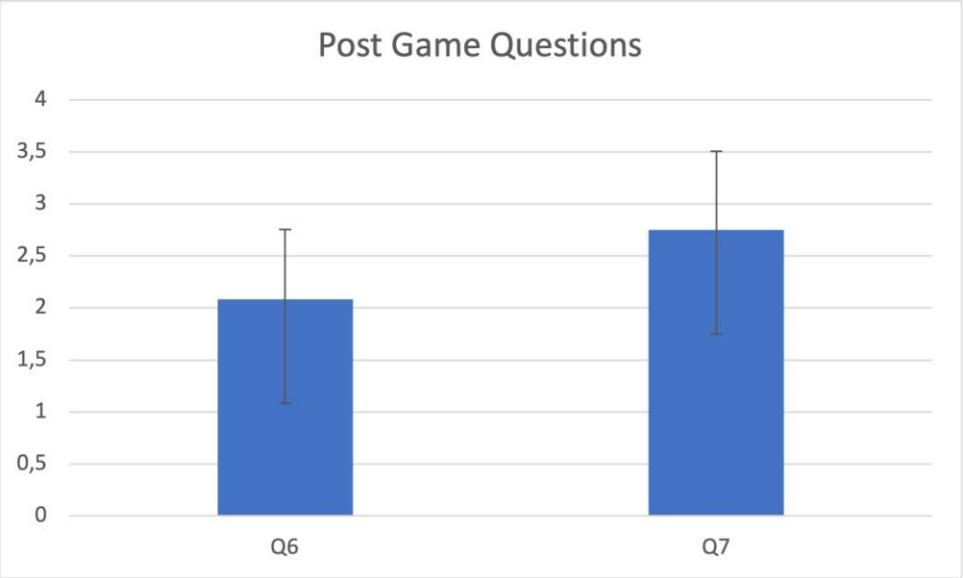

Figure 5.2: Post Game Questions

follow. In particular that the content of the terminal changes with levels. Educational information should be better highlighted. Issues with font scaling were mentioned twice. The author thought the idea concerning additional information being displayed on the right side of the screen to be a great idea. This would also deal with the issue of educational information not being highlighted enough.

To summarise, players were able to follow the game. However, multiple points of criticism were mentioned. Empty space in the view should be utilised. The game content quality should be increased in general. This issue is most pressing for chapter EoP where players took little away. Ideally, players should take something away from every level. A solution for this would be to increase level length and include more content in levels. Another solution would be to highlight educational content and changes in terminal content better. Font issue scaling should also be addressed. The general structure of the game appears to be effective. However, the aforementioned issues need to be resolved in order to make the entirety of the game effective.

5.3 Playtesting Sessions

The playtesting sessions took place in the library of the HCI Group at TU Vienna [33] on 17/05/2023 between 8:45 and 12:45. Every playtester had their own session. One session was scheduled to take 30min with a 10min buffer window in between sessions. The game was played in its locally deployed version on the author's computer. The session was screen- and audio-recorded. All sessions followed the same structure. Before playing the game the testers were asked about their computer security background. During play the

testers made use of the *thinking aloud* method to explain their thinking process. After play the testers were asked to give a review of their impressions and which information they retained best. They were also be asked about possible improvements to the game.

Playtester One had a bit of security knowledge. They were interested in the subject but felt like they stopped being able to keep up. They completed one security course several years ago. Whilst playing, they read every prompted text out loud. In the introduction view they immediately picked up on the term *HELP*. In the Tutorial the first action they did was enter *HELP* in the terminal. This triggered the task completion sound which they did not register. The playtester then tried several commands but did not know what to do in order to advance in the game. They listed all terms with the command *LIST TERMS* and then tried to get information about a term by simply entering the word into the terminal - forgetting that they would need to use a specific command to trigger this. They needed a hint to hint at the task list on the bottom right of the view. The playtester said they never registered the small text fields and would expect the view to be more prominent. They suggested to add a post-it-like field to the centre right of the screen which is a similar suggestion to one that was made in the questionnaire responses, see 5.2.2.

In the next chapter Ransomware, the playtester started checking the tasks and scrolled up in the terminal to read the commands listed in *HELP*. After not finding any useful commands, they were given a hint that they should try to enter *HELP* another time. This enabled them to make progress in the game. The playtester suggested that the changes in commands should be highlighted to the player in a way. The next section consisted of the playtester having to enable the firewall and the honeypot file. It was not obvious that all check boxes had to be checked. They assumed they had already completed the section when they had not. This section could only be completed with external hints. The last part of the level Ransomware consisted of the mini-game. There the player did not understand what the game was about. They blocked random requests without reading their descriptions. They also did not notice the text field on top of the window displaying points and misidentified requests. They failed the mini-game twice. The author explained the general idea behind the mini-game and gave them the solution. The playtester argued that too much was going on and they could not keep up. This could be solved by better explaining the context of the mini-game.

In the chapter Social Engineering they were able to follow the tasks. They entered *HELP* as the first command and could execute all tasks. However, at the intersection of last task and mini-game they were stuck. They did not know how to complete the last task which was based on starting the mini-game. Only with a hint they realised that they had to press the *Activity* button to start the mini-game. They then realised the general structure of the game: first text tasks followed by a mini-game. The playtester added a remark that this should be more obvious for the player. The remaining part of game was executed with hardly any additional remarks or issues. In the mini-game of Social Engineering they remarked that the font size was small and font spacing was sometimes squished. Every level was started with executing *HELP* followed by completing tasks

and then completing the mini-games.

The playtester tried to explore almost all text options the game provided to them and was captivated by the story. They remarked several aspects post play. As already discussed, the placement of the task list should be reevaluated. Specific terms - e.g. IDS - should be mentioned more often so that the player is made more aware of them. They also pointed out that they needed some time to get into the game and understand the game-loop. The first level proved to be quite challenging so they welcomed the decreased difficulty in the following levels. The first level should guide the player better in completing tasks. In order to achieve this, the Tutorial structure could be expanded. The explanations of the mini-games also need to be improved. They appreciated that not every mini-game was as tense as the mini-game featured in the level Ransomware. To the question whether they think the game is effective for teaching computer security, they replied with *Yes*. They found many parts of the game to be enjoyable as well as educational. In particular the first level with the setup of the firewall. They mentioned the game to be very immersive with its storyline.

The second playtester was an computer security tutor at TU Vienna. They very fast with progressing in the game. They started the Tutorial by entering *HELP*. However, they did not know how to continue. They ended up accidentally completing the Tutorial without being aware of the task list. In the chapter Ransomware they executed the first task *CREATE FIREWALL* immediately without checking command *HELP* first. After failing on the next command, they entered *HELP*. The playtester was then able to immediately enter the correct command. The follow up tasks would revolve around opening *Network Activities* and enabling the firewall and honeypot. Instead, the playtester focused on attempting to complete the tasks in the terminal. They tried opening the tab *Activity* followed by another *CREATE FIREWALL* input to no avail. They also executed *LIST TERMS* followed by a term part of the list but were unable to successfully execute it. Only later they realised that they would have to use the command *EXPLAIN <TERM>*. After external input, the playtester found the tab *Firewall*. The firewall and honeypot section was completed rapidly. They interacted with the *Activity* tab to trigger the mini-game next. In the beginning they were uncertain of how to interact with the different components of the mini-game but they eventually understood it and managed to complete the mini-game first try. They sped through the remainder of the game, only mentioning that the font size and spacing in the game of Social Engineering made it very difficult to read. They immediately knew which commands to enter to advance in the tasks. They pressed the up key and expected to see previously entered command in the terminal - just like in a normal terminal - and were surprised when that did nothing. They managed to complete the game the quickest out of all playtesters.

The playtester was very quick with picking up all the game elements. This was to be expected as they have a strong background in security. They tried to solve everything in the terminal first and had trouble finding the task list. It was also noted that the playtester hardly noticed the level transition animations as they were focused on reading the new text in the terminal. Ideally, the player should also put focus on the animations.

They remarked that some of the commands - in particular *EXPLAIN <TERM>* - are not obvious at first. Just like the previous playtester, they also mentioned that they would advise the task bar to be moved or made bigger as they did not see it at first. Also, the font size should be increased as some sections - e.g. the introductory view - were difficult to read. In their opinion, the game gamifies computer security well. It highlights several important aspects of security. However, the first level might prove overwhelming to players completely new to the subject, especially the mini-game. The game should have a slower start. As a last remark, they added that the game should introduce the main game loop better to the player as it is not clear when a task should be completed in the terminal and when it needs to be completed elsewhere.

The third playtester had only completed one security course several years ago and said they did not have a strong security background. Similar to the second playtester, they progressed very quickly through the game. The first terminal command they entered was *HELP*. Initially, they were not aware of the tasks list and did not know how to proceed. In the level Ransomware they entered *HELP* another time, letting them see the newly added commands. They entered *LIST TERMS* but did not execute the command *EXPLAIN <TERM>*. They completed the text based commands easily and initially thought the follow up tasks would have to be completed in the terminal as well. Identical to the previous playtester, they tried to prompt the previously entered command by pressing the up key and were surprised to see it did not work. After a hint, they were able to find the overlays for the firewall and honeypot setups. In the setups they did not understand at first that they had to enable every single checkbox. They thought they would only have to select specific options. In the mini-game they immediately picked up on the main-target and what to look out for. They noted that they did not understand the level's point system at first. In the beginning stage of the next level DDoS, they focused on the changes in the terminal. As a result, they did not see the animations that had been faded in. They executed *HELP* as their first action and were able to quickly complete the tasks. The playtester noted that they were not completely confident with the game's main loop but after a short moment realised that every level follows the same structure. The next level Social Engineering followed a same structure as the previous level. They completed the terminal tasks quickly. In the mini-game they noticed a bug: When interacting with the same card twice it disappears. This leads to the player having to restart the game. Apart from that, they did not immediately pick up that it was a memory game. They needed a hint of how the game works. The level EoP was completed without any additional remarks.

The playtester seemed to rush through the level and did not read the text thoroughly. They said they would expect the game to give more direct background information regarding why certain tasks have to be done. The game offers commands which give more background information but they were not executed by the playtester. In particular, they noted that they would expect levels EoP and DDoS to contain more context in order for the mini-games to make more sense. They remembered the mini-games but not the level content. For the level EoP this implies that the transition between levels

Social Engineering and EoP was not clear and that the message of the memory game was not obvious. They did not understand why the game presented the player with two open cards at the end of the mini-game. The playtester noted that the beginning was confusing as they did not immediately spot the task bar. Also, the main-game loop was not obvious at first. This was referring to tasks that cannot be completed in the terminal. Regarding the question whether they took something away from playing the game, they agreed, saying they took the most away from the first level Ransomware with the setup of the firewall.

The fourth playtester had only completed one mandatory computer security course. They were interested in additional courses but had never completed them. Upon starting the game they immediately made positive remarks regarding the humour of the game. In the Tutorial they were the first playtester to spot the task bar on the bottom right. They first executed the command *HELP*. They were then unsure of how to complete the task *Open and Close Network Tab*. They tried to execute several other commands in the terminal before being given the hint that they should click on the *Network Activities* button. They noted that they could not finish reading the text prompt in the terminal before the transition to the next level as the transition happened to quickly. They started the level Ransomware by scrolling up on the terminal to check for other commands. The playtester executed commands *EXPLAIN <TERM>* to get more information on the terms firewall and honeypot. A hint was necessary for the playtester to execute *HELP* another time to learn about level specific commands. In the setup of the firewall they did not realise that all check boxes needed to be ticked and required external input. After completing all terminal based tasks, they did not know how to proceed. They needed a hint to understand that they had to interact with the *Activities* button to trigger the mini-game. The playtester immediately picked up on the mini-game and managed it first try. They mentioned how they understood the main game loop of terminal commands followed by a mini-game.

In the level DDoS they executed *HELP* as their first command and managed to complete all tasks first try, including the mini-game. The same could be said for the level Social Engineering. However, they noted that the font was difficult to read and they would have liked the matches to be shown longer. However, they understood the switch between level Social Engineering and EoP. EoP was finished without any additional remarks.

The playtester loved the text based approach as well as the humour of the game. They felt very immersed into the game and enjoyed the game's graphical assets. They enjoyed the mini-games a lot as they were a break from the terminal's "Network security approach". They remarked that they enjoyed the Social Engineering mini-game the least as they tended to click on random combinations instead of paying attention to the content of the cards. The playtester suggested to show the matches at the completion of the level for players to understand the context of the matches better. The playtester liked the execution of the game and thought that newcomers to the security world would definitely take something away from playing the game.

The fifth playtester was an Informatics Didactic professor. They had a strong security

background. The first command they executed was *HELP*. The playtester saw the task list immediately and was able to quickly complete the Tutorial. They noted that the transition between Tutorial and Ransomware was too abrupt. The playtester also did not notice the animation on the right side of the screen just after the level transition. The text added to the terminal could not be finished reading before the level transition. In the next level they knew to execute command *HELP* to finish the terminal related tasks. However, they were confused by the reference of the terms *cups* in the level's backstory. Only after external input they understood the reference. They suggested to add more explanations for the main enemy. The next steps revolved around setting up the firewall and honeypot. They had no issues finding the setup options. However, they noted that they would expect the pop-up windows to close upon save. The playtester found the tab to trigger the mini-game quickly but said that is not clear to the player. They managed to complete the mini-game first try but noted that the game could be very confusing to players as a lot of things are going on which are not explained. The playtester sped through the rest of the game. They started every level by executing *HELP*, executing the necessary tasks and completing the mini-games. Interestingly, in the level DDoS they interacted with the *Network Activities* tab first to get any hints on what they need to do next. In the level Social Engineering they noted that the font size of the cards featured in the mini-game made it very difficult to read.

In the feedback they mentioned that they noticed they often were unsure if they needed to interact with the terminal or with the other game elements. Meaning, the mixture of terminal and other elements was not clearly separated. The playtester mentioned that they clicked on the *Network Activities* tab to understand how to progress in the level. Buttons being disabled in this tab left them confused several times whilst playing the game. They suggested to introduce better explanation of the different game focus points. They also noted that pop-ups appeared on top of one another which could be confusing or annoying for players. They also noted that the windows should close themselves when pressing *Save*. The latter is only relevant for the honeypot pop-up window but a *Save* button could also prove useful for the firewall pop-up window. When prompted with the question whether they think players might take something away from playing the game, they said that the first mini-game might be confusing to players as they themselves did not know what to do at first. The other parts of the game were liked by the playtester. In particular they enjoyed that certain terms developed whilst continuing the storyline, i.e. the term IDS. The term is first part of the storyline in level DDoS and again part of the storyline in Social Engineering with extended explanations and references. Also they noted that the example given in Social Engineering could have been phrased in a clearer manner in order for players to fully understand the importance of the topic.

The sixth playtester had completed one mandatory security course. They mentioned in the Introduction section that the font size was very small which made it more difficult to read. The first task they executed in the terminal was *HELP*. They then tried different commands to see which effects they might have. They tried *GRAB COFFEE*, *LIST TERMS* and *MIN*. They needed an external hint on the location of the task list. After

this hint, they quickly finished the Tutorial. However, they remarked that the transition between the Tutorial and level Ransomware was too fast. They were unable to finish reading the added text. In the level Ransomware they needed input to learn that the *HELP* command contains new commands. The playtester then executed *HELP* followed by *LISTEN REQUESTS*. They opened the *Network Activities* tab where they realised they needed to execute *CREATE FIREWALL* in the terminal. They noted that it is not obvious when to interact with the terminal and when to interact with other parts of the game. The playtester then immediately knew they needed to open the *Network Activities* tab followed by the *Firewall* tab to finish the setup of the firewall. They knew that they would have to enable all check boxes to complete the setup. After this, they quickly found their way to the mini-game. The mini-game was not obvious to the playtester. They blocked random requests and needed to retry the mini-game twice. After an explanation they understood the concept. They argued it was not obvious to them where to look. Level DDoS was started with the command *HELP* and all tasks were quickly completed. The same could be said for levels Social Engineering and EoP. For the mini-game in the level Social Engineering they noted that they were lacking context and did not know what they were doing. They would appreciate a more detailed explanation.

In the feedback the playtester noted that the game reminded them of the game *Hacknet* [34]. The author had not heard about the game before. The playtester noted that they did not understand the motivation behind the mini-game in Social Engineering and the corresponding connection to EoP. They mentioned that they had not heard about honeypots before and were now interested in the setup of one. They would have liked a more detailed explanation of that term. At the end of the conversation they mentioned they would potentially be interested in taking another non-mandatory security course.

To summarise, they playtesting was an effective way to evaluate game issues and receive game feedback. Based on the player feedback it can be said that the game is already effective in conveying its core information to players. Playtesters enjoyed the humour, game assets and general game structure. All playtesters either took something away from the game or could see how other players could take something away. A lot of points of criticism which were mentioned in the playtesting session were also mentioned in the player questionnaires. Given the qualitative nature of playtesting, the issues were easier to understand. Many points were mentioned which would need improvement. Notably, there were the location of the task list, the transition between terminal and other game elements, font size and font spacing, the context of several game parts - i.e. terms, level transitions and mini-games - and the growing number of commands added with every level. Players did not explore the terminal commands but simply wanted to complete the tasks as fast as possible. This lead to several commands not being executed at all, i.e. *LIST TERMS* in any chapter apart from the Tutorial. Also, player focus must be shifted correctly. Players should notice the animations which are displayed in the main window. This did not happen on several occasions during the playtesting because playtesters were too focused on the terminal. The aforementioned issues must be resolved for the game to increase its game design qualities and overall effectiveness in knowledge retention.

5.4 Game Improvements

5.4.1 Game Bugs

Several game bugs were mentioned in the questionnaire answers and playtesting session feedback. One of the most pressing issues that was mentioned many times was the font size and font spacing. It was noted several times that the font size of several portions of the game was too small to read with ease, notably in the introduction section of the game and in the task list. Increasing the font size would increase readability. However, the more pressing font issue is the font spacing. In some parts of the game the text letters appear to be cut off and not rendered correctly, see also 5.3. Players explicitly mentioned not being able to properly read text at all due to this issue. The issue occurs 100% of

Figure 5.3: Font Issues

time in the mini-game of the level Social Engineering but it can also be reproduced in the terminal when adjusting the window size. The author fixed a similar issue in the development phase already where the text in the terminal would clip a lot leading to the text to be barely readable. This was fixed by increasing the bottom text margin. In the case of the scaling issues in the mini-game 5.3 and scaling issue, the issue is related to spacing between characters. The spacing between characters should be increased. It should be noted that this change could lead to unwanted side-effects. Notably, characters not being close enough to characters that are part of the same word. As changing the font spacing would have an effect on the entire game, this would need to be thoroughly tested - making it a potentially very time consuming fix. However, the issue needs to be fixed as people should be able to resize the game window without sacrificing font readability. The fix might prove complex.

The mini-game of the level Social Engineering contains two known bugs. The first issue is that the same card can be clicked twice and then disappears, making the player unable to finish the level. The second issue revolves around the level not restarting properly. Upon a mini-game restart, only two cards are shown instead of the expected eight leading the player to be stuck in that part of the game. Fixing these two issues is straight forward

as they were caused by simple overlooks. For issue one, an additional logical statement needs to be added. For issue two, a variable manipulating the size of the displayed cards had a reference a static variable holding information to all cards. The variable manipulating the size should instead copy the static variable and then manipulate the size. As these two bugs were straight forward, they were fixed right after being made aware of them after the playtesting sessions. The mini-game EoP contains one known bug. If the player falls off the cliff at the last section of the game, they are not prompted with a game over screen. This was already fixed during the questionnaire phase. For the issue to be fixed, the size of a collision element needed to be increased.

5.4.2 Game Content

Based on the results of the questionnaires and playtesting, the game content is effective in conveying its messages. However, several players pointed out that some levels were too short. This was noted about levels DDoS and EoP in particular. Additional tasks would have to be added. For level EoP a potential additional task would be *DELETE USER JULIA*. Prior to that task players would have to scan the system again to find the malicious user. If the player paid attention in the mini-game of Social Engineering, they would know the user immediately, making them skip this step. For level DDoS an approach would be to make the checking of the server capacity more engaging. This would imply that the player had to find the concrete server that was being attacked by utilising newly introduced commands. For level Social Engineering it was noted that the level should highlight the importance of the topic. This could be achieved by mentioning the storyline more often in the terminal updates and to highlight it with a longer introduction in the terminal.

Apart from player feedback, more request variety for level Ransomware should be implemented as well as additional memory cards in the mini-game featured in level Social Engineering. This is feasible to implement.

5.4.3 Game Design

Several game design issues were noted by players whilst playing the game. The most apparent design flaw is the location of the task list. Its location needs to be moved and its size needs to be increased to make it easily visible. It was mentioned multiple times that the location of the list should be on the top to centre right of the screen. This would also solve the issue of the right side of the main screen being empty. Players started the Tutorial level blind not knowing what to do because they could not locate the task list. To solve this, the list needs to be moved, as suggested by players, see 5.4. Its size should be increased and its location should be moved. The font size of the tasks should also be increased. Players should be immediately able to spot the list. This solution would be straight forward to implement. It should be noted that the location is identical to the location of the level transition animations. The task list would have to be hidden in the level transition phase.

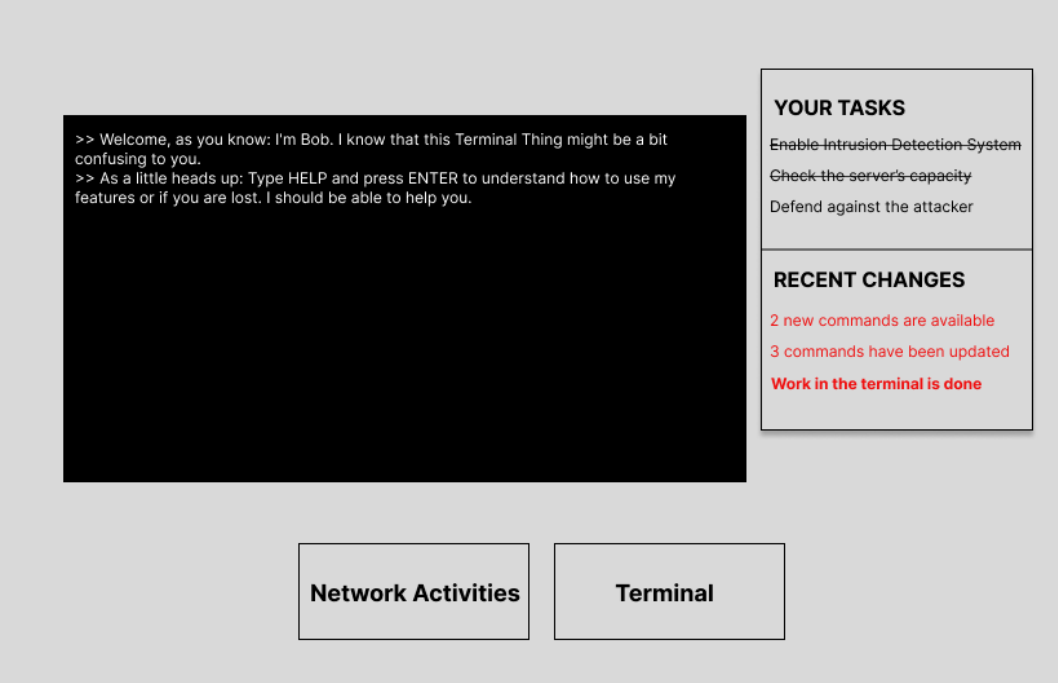

Figure 5.4: Moved Task List

The transition between terminal and other game elements should be better explained to the player. Many times players tried to complete tasks in the terminal which were actually about completing the mini-game. A related issue was that players did not understand that the commands listed by *HELP* expanded as the game progressed. They also had no incentive to explore different commands. A solution to this would be to feature *game updates* in the main desktop. As already shown in 5.4, the task list could contain a subsection with recent changes. The recent changes section would feature newly, changed commands or interesting commands. It would also guide the player to where they would have to shift their focus to. The section would include a text based remark guiding the player to whether focus needs to be shifted to the terminal or to other game elements. The section could be seen as an extension of the terminal, supporting the player in continuing in the story. The transition between terminal and other game elements could also be explained via a text prompt in the terminal. The chosen solution would be based on player feedback. Identical to the task list change, these changes would be feasible to implement. However, they would be more complex as many different cases would have to be considered.

Another game design issue is related to the player introduction to the mini-games. Players very often started the mini-games without thoroughly reading the explanatory text provided right before triggering the mini-game. This lead to them being confused and not understanding the context of the mini-game. A solution to this would be to introduce a tutorial overlay to the mini-game right after triggering the game. It would

describe every single game element and its purpose by showing text combined with arrows pointing to the corresponding game elements. It would also explain the winning and losing conditions to the player. This would enable players to easily understand the game prior to actually starting it. As an example, the mini-game of the level Ransomware would explain the incoming requests section, arrows and the filter sections, amongst several other game elements, see 5.5. This change should be straight forward to implement.

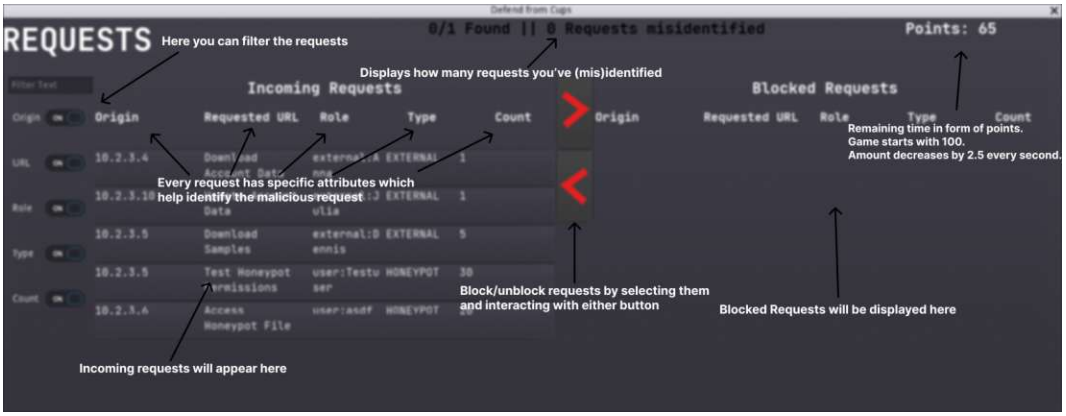

Figure 5.5: Tutorial Ransomware Mini-Game

Many times players did not notice the level transition animations because they were focused on the newly added text in the terminal. A solution to this would be to place the animations on top of the terminal. Once the animation has completed, the animation would fade out and the terminal could display its updated text. This way players would not have to focus on two different parts of the view. This would also solve the issue of the conflicting locations of the updated task list 5.4 and the level transition animations. This change is feasible to implement.

The point systems of the different mini-games need to be revamped. Players were many times left confused as to what they meant and ignored them. Points should be either better embedded into the game or removed entirely.

The firewall and honeypot pop-up windows should convey to the player that all check marks have to be checked. Players thought they had to choose between the options. Also, a *Save* button should be added to the firewall pop-up. Interacting with the *Save* button should close the pop-up window.

Two playtesters attempted to press up on the terminal to prompt the previously entered command as this can be done in any real terminal. Implementing this feature for the game terminal would make it feel more realistic and the implementation would be straight forward.

All noted game design changes are feasible and straight forward to implement. Implementing them would drastically elevate the gameplay experience. However, future improvements to the game such as more content and improved visual design of levels need

to be considered as well to make the game more appealing. To evaluate the effectiveness of the presented changes, a follow-up playtesting session would have to conducted. This is outside the scope of this thesis.

CHAPTER 6

Conclusion & Outlook

In the context of this thesis a game prototype was developed which had as its aim to teach computer security to computer science students who do not have a strong security background. The game idea was based on a novel idea from the author and was improved upon in several design iterations. The game features four levels each focusing on one computer security topic: Ransomware, DDoS, Social Engineering and EoP. The main game-loop revolves around text interactions in a terminal-like element followed by a mini-game revolving around the level's topic. The game was evaluated using questionnaires and playtesting sessions. The goal of the evaluations was to measure the effectiveness of the game design and content and to spot any potential design flaws.

Players needed to complete two questionnaires, one pre- and one post-play. The questions focused on understanding knowledge and interest pre- and post-play. They were both single-choice and open-text questions. The questionnaires yielded positive results. Most players took something away from playing the game and could remember information presented in the game. The quality of knowledge specific answers remained mostly the same despite participants having less computer security knowledge on average post-play. Interest in the field was also increased. Several remarks were made regarding game improvements. In particular, it was noted that levels should feature more content and that the game should scale correctly when resized.

Playtesting sessions were focused on understanding player interaction with the terminal and detecting any potential game design issues. Several game design issues were discovered. On some occasions, it was not clear to the player whether they had to interact with the game or interact with other game elements. The location of the task list - which is used to track player progression - was also not immediately apparent to players. Similar to the feedback given in the player questionnaires, playtesters would have liked levels to be longer and contained more content. It was also noted that more background information should be given on specific topics and mini-games. However, all playtesters noted that

they either took something away or could see how players could take something away from playing.

Based on the results of the questionnaires and playtesting sessions, the prototype was a success. The game's current version successfully conveyed its main points to players and increased interest in the field of computer security.

For future iterations, game design and content issues should be resolved. Game length should be increased, more context should be given on game content and players should be better guided to progress in the storyline.

List of Figures

List of Tables

Acronyms

CTF Capture the flag. 33

DDoS Distributed Denial-of-Service Attack. 13–15, 18, 21, 22, 28, 38, 42, 46–49, 51, 55, 57

EoP Elevation of Privileges. 13, 14, 16, 18, 21, 22, 29, 31, 37, 39, 40, 42, 43, 46, 47, 49, 51, 55, 57

IDS Intrusion Detection System. 13, 14, 22, 28, 29, 37, 39, 45, 48

IPS Intrusion Prevention System. 14, 22, 28

NPC Non-Playable Character. 15

Bibliography

[1] *Denkweisen der Informatik*. [Online]. Available: https://www.tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=187B12&semester=2022W&dswid=6871&dsrid=393 (visited on 10/07/2022).

[2] *Aurora*. [Online]. Available: https://aurora.iguw.tuwien.ac.at/course/dwi/login/?next=/course/dwi/ (visited on 10/07/2022).

[3] L. M. Caldas, H. N. Eukel, A. T. Matulewicz, E. V. Fernández, and K. L. Donohoe, „Applying educational gaming success to a nonsterile compounding escape room“, en, *Currents in Pharmacy Teaching and Learning*, vol. 11, no. 10, pp. 1049–1054, Oct. 2019, ISSN: 1877-1297. DOI: 10.1016/j.cptl.2019.06.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877129718304593 (visited on 10/08/2022).

[4] *Dekanatsteam*, de. [Online]. Available: https://informatik.univie.ac.at/fakultaet/leitung/dekanatsteam/person/20965/ (visited on 10/09/2022).

[5] *Research Unit Security and Privacy*, en. [Online]. Available: https://informatics.tuwien.ac.at/orgs/e192-06 (visited on 10/09/2022).

[6] G. Engine, *Godot Engine - Free and open source 2D and 3D game engine*, en. [Online]. Available: https://godotengine.org/ (visited on 10/08/2022).

[7] D. Capello, *Aseprite*, en. [Online]. Available: https://www.aseprite.org/ (visited on 10/09/2022).

[8] *Download the latest indie games*, en. [Online]. Available: https://itch.io/ (visited on 10/09/2022).

[9] *CryptPad: Collaboration suite, encrypted and open-source*. [Online]. Available: https://pads.c3w.at/ (visited on 03/29/2023).

[10] G. Blakely, H. Skirton, S. Cooper, P. Allum, and P. Nelmes, „Educational gaming in the health sciences: Systematic review“, en, *Journal of Advanced Nursing*, vol. 65, no. 2, pp. 259–269, Feb. 2009, ISSN: 03092402, 13652648. DOI: 10.1111/j.1365-2648.2008.04843.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2648.2008.04843.x (visited on 10/07/2022).

[11] C. Neustaedter, „Analysis of Gamification in Education“, en,

[12] J. Vykopal and M. Barták, „On the Design of Security Games: From Frustrating to Engaging Learning“, en, 2016. [Online]. Available: https://www.usenix.org/conference/ase16/workshop-program/presentation/vykopal (visited on 10/08/2022).

[13] K. Boopathi, S. Sreejith, and A. Bithin, „Learning Cyber Security Through Gamification“, *Indian Journal of Science and Technology*, vol. 8, no. 7, pp. 642–649, Apr. 2015, ISSN: 0974-5645, 0974-6846. DOI: 10.17485/ijst/2015/v8i7/67760. [Online]. Available: https://indjst.org/articles/learning-cyber-security-through-gamification (visited on 10/08/2022).

[14] M. Beltrán, M. Calvo, and S. González, „Experiences Using Capture The Flag Competitions to Introduce Gamification in Undergraduate Computer Security Labs“, in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2018, pp. 574–579. DOI: 10.1109/CSCI46756.2018.00116.

[15] Z. C. Schreuders and E. Butterfield, „Gamification for Teaching and Learning Computer Security in Higher Education“, en, 2016. [Online]. Available: https://www.usenix.org/conference/ase16/workshop-program/presentation/schreuders (visited on 10/08/2022).

[16] R. Roepke and U. Schroeder, „The Problem with Teaching Defence against the Dark Arts: A Review of Game-based Learning Applications and Serious Games for Cyber Security Education:“ in *Proceedings of the 11th International Conference on Computer Supported Education*, Heraklion, Crete, Greece: SCITEPRESS - Science and Technology Publications, 2019, pp. 58–66, ISBN: 9789897583674. DOI: 10.5220/0007706100580066. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007706100580066 (visited on 10/08/2022).

[17] A. Yasin, L. Liu, T. Li, R. Fatima, and W. Jianmin, „Improving software security awareness using a serious game“, en, *IET Software*, vol. 13, no. 2, pp. 159–169, 2019, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-sen.2018.5095, ISSN: 1751-8814. DOI: 10.1049/iet-sen.2018.5095. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2018.5095 (visited on 06/12/2023).

[18] S. Hart, A. Margheri, F. Paci, and V. Sassone, „Riskio: A Serious Game for Cyber Security Awareness and Education“, en, *Computers & Security*, vol. 95, p. 101 827, Aug. 2020, ISSN: 0167-4048. DOI: 10.1016/j.cose.2020.101827. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820301012 (visited on 06/12/2023).

[19] S. Frey, A. Rashid, P. Anthonysamy, M. Pinto-Albuquerque, and S. A. Naqvi, „The Good, the Bad and the Ugly: A Study of Security Decisions in a Cyber-Physical Systems Game“, *IEEE Transactions on Software Engineering*, vol. 45, no. 5, pp. 521–536, May 2019, Conference Name: IEEE Transactions on Software Engineering, ISSN: 1939-3520. DOI: 10.1109/TSE.2017.2782813.

[20] M. Mostafa and O. S. Faragallah, „Development of Serious Games for Teaching Information Security Courses“, *IEEE Access*, vol. 7, pp. 169 293–169 305, 2019, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2955639.

[21] *TryHackMe | Cyber Security Training*. [Online]. Available: https://tryhackme.com/hacktivities (visited on 05/11/2023).

[22] *ThreatGEN: Red vs. Blue on Steam*, en. [Online]. Available: https://store.steampowered.com/app/994670/ThreatGEN_Red_vs_Blue/ (visited on 05/16/2023).

[23] *Uplink on Steam*, en. [Online]. Available: https://store.steampowered.com/app/1510/Uplink/ (visited on 05/16/2023).

[24] *A Normal Lost Phone on Steam*, en. [Online]. Available: https://store.steampowered.com/app/523210/A_Normal_Lost_Phone/ (visited on 05/16/2023).

[25] *(08/12) Casey Joint by Yahtzee Croshaw*, en. [Online]. Available: https://yzcroshaw.itch.io/casey-joint (visited on 06/05/2023).

[26] *OWASP Top Ten | OWASP Foundation*, en. [Online]. Available: https://owasp.org/www-project-top-ten/ (visited on 10/07/2022).

[27] R. Vishwakarma and A. K. Jain, „A survey of DDoS attacking techniques and defence mechanisms in the IoT network“, en, *Telecommunication Systems*, vol. 73, no. 1, pp. 3–25, Jan. 2020, ISSN: 1572-9451. DOI: 10.1007/s11235-019-00599-z. [Online]. Available: https://doi.org/10.1007/s11235-019-00599-z (visited on 05/16/2023).

[28] Z. Wang, L. Sun, and H. Zhu, „Defining Social Engineering in Cybersecurity“, *IEEE Access*, vol. 8, pp. 85 094–85 115, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2992807.

[29] *Free Pixel-art background, desert (Day/Night) by blank_canvas*, en. [Online]. Available: https://blank-canvas.itch.io/parallax-pixel-art-background-desert (visited on 05/16/2023).

[30] *DiffusionBee - Stable Diffusion App for AI Art*. [Online]. Available: https://diffusionbee.com/ (visited on 05/16/2023).

[31] *JetBrains Mono: A free and open source typeface for developers*, en. [Online]. Available: https://www.jetbrains.com/lp/mono (visited on 05/19/2023).

[32] *Juliastic/Project-Secure*, en. [Online]. Available: https://github.com/juliastic/Project-Secure (visited on 05/17/2023).

[33] *Homepage IGW - HCI | Human Computer Interaction Group*. [Online]. Available: http://igw.tuwien.ac.at/hci/ (visited on 05/16/2023).

[34] *Hacknet on Steam*. [Online]. Available: https://store.steampowered.com/app/365450/Hacknet/ (visited on 05/18/2023).