Vulcan - The video game to teach Informatics students about Firewalls

Rusab Abrez Asher

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2019

Abstract

Firewalls play a fundamental role in Computer Security, but configuring the rules for them may prove challenging to beginners in Computer Security. This project discusses the implementation and evaluation of a game to help players understand how firewalls work. It teaches users about firewall rules and how information is allowed or denied. The target audience of this project is second, third and fourth-year students who have not taken a course on Computer Security or wish to learn more about how to configure rules in a firewall.

The game teaches how to set up firewall rules using the Linux firewall command line tool `iptables`. It teaches the general syntax of this tool, as well as any information required to understand the workings of firewalls at a moderate level. It consists of eight levels, with each level targeting a different aspect of firewall configuration. The information from previous levels is built-upon in later levels.

My approach consists of building an initial design, then changing this design from additional research, resulting in a new game. The game is then tested on the target audience to evaluate if it reaches its learning goals and is a usable system.

Acknowledgements

I want to thank everyone who has been a part of the evaluation of this project.

Without the guidance of my supervisor, Kami Vaniea, I would not have been able to get as far as I did, so I would like to thank her humbly.

I would also like to thank my friend Stephen for assisting me throughout this project in understanding different ways of evaluation as well as proofreading my thesis several times.

Last but not least, I'd like to thank both my family and friends who have been supportive throughout my University years.

Contents

1 Introduction **9**
1.1 Project Aim 9
1.2 Report Structure 9

2 Firewalls and their usage **11**
2.1 What are Firewalls 11
2.2 The OSI Model 11
2.3 Types of Firewalls 13
2.4 The Importance of Firewalls 13
2.5 Protocols 13
2.6 iptables 14
2.7 Limitations of Firewalls 16
2.8 Summary 17

3 Educational Games **19**
3.1 Games for education 19
3.2 Security Games 20
3.2.1 Zachtronics 20
3.2.2 Permission Impossible 21
3.2.3 Blue Team 21
3.3 Summary 22

4 Game Design **23**
4.1 Technologies Used 23
4.1.1 Programming Language and IDE 23
4.1.2 Graphical Design 24
4.1.3 Version Control 24
4.1.4 Other Game Assets 24
4.2 First Design 25
4.2.1 Concept 25
4.2.2 Intentions 25
4.2.3 Story Design 25
4.2.4 User Interface Design 26
4.2.5 Game Analysis 28
4.2.6 Learning Outcomes 28
4.3 Second Design 29

4.3.1 Concept . . . 29
4.3.2 Requirements . . . 29
4.3.3 User Learning Goals . . . 31
4.4 Summary . . . 32

5 Game Implementation 33
5.1 Iterative Development . . . 33
5.2 Game layout . . . 33
5.3 Game Scenes . . . 34
5.3.1 Level Design . . . 37
5.4 Game Statistics . . . 40
5.4.1 Lines of Code . . . 40
5.4.2 Build Size . . . 40
5.4.3 Resolution . . . 40
5.4.4 Memory Usage . . . 40
5.4.5 CPU Usage . . . 41
5.5 System Requirements . . . 42
5.6 Summary . . . 42

6 Evaluation Methodology 45
6.1 Pre-game questionnaire . . . 45
6.1.1 Section 1 . . . 45
6.1.2 Section 2 . . . 45
6.2 Think-Aloud . . . 46
6.3 Post-game questionnaire . . . 46
6.3.1 Section 1 . . . 46
6.3.2 Section 2 . . . 47
6.4 Experiment Details . . . 47
6.5 System Usability Scale . . . 47
6.6 Summary . . . 48

7 Results and Evaluation 49
7.1 Iteration . . . 49
7.2 First Iteration . . . 49
7.2.1 Game Design . . . 49
7.2.2 Evaluation with Experts . . . 50
7.2.3 Conclusion of Expert Evaluation . . . 52
7.3 Second Iteration . . . 53
7.3.1 Results . . . 53
7.4 Evaluation Conclusion . . . 59
7.5 Summary . . . 60

8 Conclusion 61
8.1 Overview . . . 61
8.2 Results achieved . . . 62
8.3 Future Work . . . 62

A Game Images **63**

B Requirements Gathering Google Forms Questionnaire **67**

C Consent Form **71**

D Pre-game Questionnaire **73**

E Post-game Questionnaire **77**

F Think-Aloud Script **81**

G SUS Survey **83**

Bibliography **85**

Chapter 1

Introduction

1.1 Project Aim

This project aims to design and implement an educational game to teach players about firewalls. The players go through different levels, each level teaching them new concepts as well as using concepts taught in preceding levels.

The target audience for the game is second, third and fourth-year undergraduate students who are studying Computer Science or a related degree. This project is intended for assisting the learning of firewall rules that are taught in the Computer Security Course by the School of Informatics at the University of Edinburgh. We set proper learning goals for the game based on feedback from students who had already taken the Computer Security course. Using these learning goals, we form requirements for the system. The game was then designed accordingly.

Using a pre-game and post-game questionnaire, we evaluated the teaching capabilities of the game to ensure that it satisfied the learning goals. To assess the usability of the game, we also used the System Usability Scale and had participants play the game in a think-aloud scenario.

1.2 Report Structure

The proceeding chapters are organised as follows:

Chapter 2: provides an overview of Firewalls and `iptables`, what their purpose is and the motivation for the project.

Chapter 3: talks about different educational games as well as games made for teaching security. We analyse what stood out for different players of these games and what can be integrated from these games into our own game.

Chapter 4: details the technologies used for this project. We also present an initial design for the game, what we changed from this design and why. We then present our second design concepts, and detail how requirements were gathered for this game, from which learning goals were derived.

Chapter 5: shows the final version of the game, the different scenes the player interacts within the game and outlines the different learning outcomes for each level. In addition, this details the system requirements in order to run the game.

Chapter 6: describes the techniques used to assess the game's effectiveness in teaching, and why these techniques were chosen to evaluate this project.

Chapter 7: evaluates the results received from the players and compares the learning goals of the game to what is actually learned after playing the game.

Chapter 8: summarises what this project has achieved. Suggestions on how this project could be improved on in the future are also discussed.

Chapter 2

Firewalls and their usage

In this chapter, we will discuss firewalls; what they are and why they are used. We will also discuss the different protocols used in networking. We will then talk about `iptables` - a command line tool for configuring firewalls on the Linux operating system.

2.1 What are Firewalls

A firewall is a network security system with the main purpose being to prevent unauthorised access to or from a private network [1]. To understand firewalls, we must first define a network packet. A packet represents the smallest amount of data that can traverse over a network at a single time[2].

Firewalls filter these packets based on many factors, some of which include where the packet is from and who it is going to. It does this by creating rules to determine if the packet is conforming to authorised activities that are considered legal and expected in the network. Firewalls are used to prevent social engineering trojans, outdated software, phishing attacks, network travelling worms and advanced persistent threats [1]. There are many different kinds of firewalls, some of which are packet filtering firewalls, application proxy firewalls, stateful inspection Firewalls and hybrids. [3]. Different firewalls prevent different threats, but packet filtering firewalls are the most basic type which blocks unauthorised access and traffic.

2.2 The OSI Model

The Open Systems Interconnection or OSI Model defines a networking framework that represents how applications communicate with one another over a network. It does this by dividing the communications in a system into seven different layers. Different firewalls operate at varying levels of the OSI model depending on their functionality. Each layer in the OSI model provides functionality and services for the layer above it. In

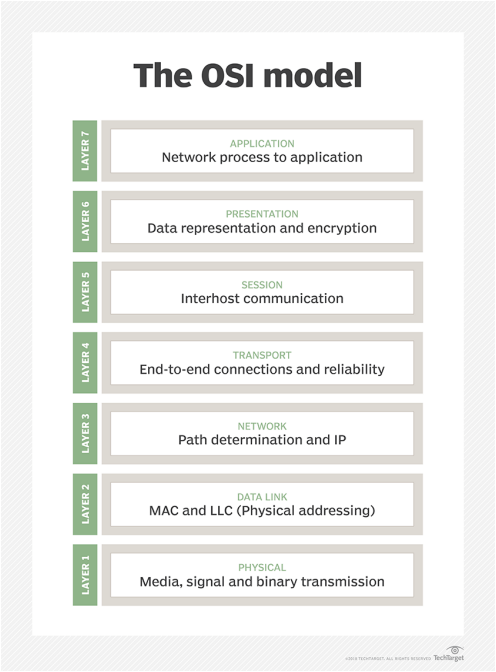


Figure 2.1: The OSI Model, taken from Search Networking [4]

the application layer, the identity of the computers involved in the communication are discovered, i.e. if there is someone to communicate with, if the network will allow them to talk right now and if the data is in a form that the user who is receiving it can understand.

The presentation layer converts the information that is going into or out of the network from one form to another, e.g. it can convert clear text to encrypted text.

The session layer is where the conversation of two or more computers is configured. This layer is also where these conversations are terminated.

The transport layer is where data is separated into packets. It also manages the delivery of these packets as well as any error checking in case a packet is lost.

The network layer handles the addressing and sending of the data i.e. making sure it goes in the right direction to the correct destination. It also handles incoming transmissions at the packet level.

The data link layer is what sets up the links in a network for communication. This layer consists of the Logical Link Control Layer, which manages the network traffic over the physical medium, and the Media Access Control or MAC Layer, which concerns sharing the physical connection of the network among several different computers.

Finally, the physical layer transports the stream of packets across the network either electrically, mechanically or through radio waves.

2.3 Types of Firewalls

There are many types of firewalls. We will briefly go over packet filtering firewalls, stateful inspection firewalls and application inspection firewalls.

Stateful inspection Firewalls monitors the state of active connections and uses this information to determine which network packets to allow through the firewall. Stateful inspection monitors the communication of packets over some specified period, examining both incoming and outgoing packets.

Application proxy firewalls are network security systems that protects network resources by filtering messages at the application layer in the OSI model. This type of firewall can also analyse the packet to determine if that packet is harmful to the network. In this project, we will discuss stateful inspection firewalls using a Linux command line utility called `iptables`.

Packet filtering firewalls control access to a network by analysing the incoming and outgoing network traffic based on where it has come from and where it is going to [5]. They are simple to implement and are used when the security requirements are modest. Packet filtering firewalls perform only elementary operations, such as examining the packet header, verifying the IP address, the port or both, and granting and denying access without making any changes [6]. This simplicity leads to efficient filtering. The filtering of packets is based on their source IP address, destination IP address, source port, destination port, and the protocol used [6].

2.4 The Importance of Firewalls

The configuration of firewalls is the most critical part of the security that a firewall can provide [7]. While vulnerabilities in firewall software itself are comparatively rare, it is commonly known that firewalls enforce poorly written rule sets, which are configured by system administrators [8]. Poorly written rule sets result in a configuration error, which can be defined in many ways depending on the context in question. One rule set for a firewall for a particular organisation might be completely unacceptable for another organisation[7].

Another reason is the massive size of some rules sets, which increases the complexity of a system, resulting in a configuration errors.[7].

2.5 Protocols

A packet has its own rules which are defined by a networking protocol. These rules determine how the packet is structured as well as the procedure took if the receiving computer does not get a packet within a certain time during its transmission. One such

protocol is the Transmission Control Protocol or TCP. This protocol is widely used in computer networking to maintain the ordering of packets when a computer sends a message. When a computer receives a packet, it sends a packet back to the sending computer called an 'Acknowledgement'. This 'Acknowledgement' is a means of confirmation for the sending computer that its packet did not get lost. If the sending computer does not receive an 'Acknowledgement' within a certain time of sending the packet, it will resend the packet to the receiver. In this way, TCP ensures no packet loss albeit at the cost of speed.

Another protocol is the User datagram protocol or UDP. This protocol does not handle packet loss in the way that TCP does, as it does not send Acknowledgements. UDP is primarily used when the speed of transmission is more important than complete packet transfer, e.g. a video conference call, where some packets can go missing if the call is kept consistent and fast.

TCP and UDP are known as internet Protocols. They provide functionality for a different type of protocols called the Application Protocols. There are many Application Protocols. One of the Application Protocols is the HyperText Transfer Protocol, or HTTP, which uses TCP to communicate. HTTP is used for viewing web pages online. Each Application protocol can also be identified by a port number, which is commonly unique to that application protocol. Another application protocol is the Secure Shell Protocol or SSH protocol. This protocol is commonly used for logging into a computer remotely from another computer.

A firewall can block or allow packets to pass through it depending on the protocol of that packet. E.g. a firewall can block Web Traffic for a network by not allowing HTTP packets through the firewall, given that all traffic to the internet for that given network pass through the firewall.

2.6 iptables

`iptables` is a user-space utility program that allows a system administrator to configure the tables provided by the Linux kernel firewall [9]. It provides an interface and command line tool to configure a firewall so that it can block or allow certain packets. The system administrator or user can use `iptables` to write out different rules that determine whether a packet should be allowed through or not based on its properties. A set of rules is known as a chain of rules, and a set of chains is known as a table.

Once a packet is matched according to its protocol and IP address, the firewall does what the policy states for that rule. A policy determines whether the firewall should allow the packet to pass through, which is defined by the **ACCEPT** word, or to not allow it through, which is defined by the **DROP** word. There is also a policy called **RETURN**, which means to skip the current chain and go back to the next rule from the chain in which it was called [10].

While the user can create their tables, the default table is called `filter`, and consists of three chains: [10]

- **INPUT:** This chain controls incoming packets to the network the firewall is on.
- **OUTPUT:** This chain controls the outgoing packets from the network it is on.
- **FORWARD:** This chain filters packets that come into your network to be routed to another network.

When a packet is either coming into the network or leaving the network, the `filter` table is looked up, and the packet is matched according to its respective chain, e.g. if it is leaving the network, it matches with the **OUTPUT** chain. If a rule is found, then that specific action is matched against that packet. Once a packet has been matched with a rule, no other rules are read for that packet. Thus, the ordering of rules is of the utmost importance.

The following is an example of what a command to make a rule looks like :

```
iptables -A INPUT -s 192.165.21.22 -p tcp --dport 80 -j ACCEPT
```

In the above example, `iptables` is the name of the command and has specific parameters. `-A` means this rule will be added to the end of a chain, although there are other ways to add a rule to a chain. After `-A`, the chain name to add to is specified, in this case, the **INPUT** chain. You can also specify a particular source IP address for the rule after `-s`.

The protocol of the rule is specified after `-p`; in this case, it is TCP. The application protocol for a rule is defined by the `--dport` parameter, which means destination port. This specifies which port the packet will go through in the receiving computer. Instead of the port number, the application protocol can also be written, which in this case would be 'http'. Lastly, we define the policy for this rule using the `-j` parameter, which in this case is to **ACCEPT** the packet. There are many other parameters in `iptables`, but we shall not go into detail about them. In the above example, if the source IP Address, the protocol or the destination port are not specified, the rule will default to ALL source IP addresses, protocols or destination ports respectively.

There are many other useful commands that `iptables` provides. Chains also have a Default Policy. This policy states what should happen to a packet if there is no rule regarding that packet in the current chain. The firewall will then either **DROP** or **ACCEPT** the packet, depending on what the default policy states. The command to change a default policy is:

```
iptables --policy INPUT DROP
```

This will change the default policy of the **INPUT** chain to **DROP**, hence not allowing a packet to pass through if there is no rule found in the **INPUT** chain matching it.

The last command we will cover is the command to delete a rule. As mentioned before, `iptables` does not provide the feature to reorder rules in its chains; thus one way to reorder rules is to delete rules and add them later on. The deletion of rules is done using the line number of that rule in its respective chain. The deletion of rules is done by the following command:

```
iptables -D INPUT 1
```

The above command will delete a rule in the INPUT chain at line number 1. The line number is the last argument in the above command.

2.7 Limitations of Firewalls

There are multiple types of firewalls catered to different purposes, and they all have their limitations. A firewall cannot defend against a threat that is within your network; hence it cannot protect against a user who abuses authorised access to your domain [6]. Another limitation is that it does not defend against viruses. The complexity of the packet filtering rule setup for a vast network can result in vulnerabilities in the defence [6].

T. Chomsiri, X. He and P. Nanda mention in their paper [11] the impact that a shadowed rule can have on the security and speed of a firewall. A shadowed rule is a rule that cannot be matched with a specific packet. That is because that packet would already be acted upon by the rules above the shadowed rule. They go on to say that a speed problem occurs when there are a lot of shadowed rules. This is due to wasted processing time on useless rules, i.e. most packets would not match all rules and go to the last rules of the firewall or the default policy, which results in low throughput. They also say that if a rule to protect against a specific attack is not in the first few lines of the rules in a chain, it could be 'shadowed' by rules above it which might allow those specific packets.

Cloudflare is a company based in the U.S. that provides content delivery network services, DDoS mitigation, internet security and distributed domain name server services. Cloudflare allows users to configure their firewalls through a user interface. Also, they provide an API for configuration of firewalls. According to the Cloudflare, one of the most significant feedback they received was that users wanted the ability to reorder firewall rules without having to delete and insert them again. [12]. Subsequently, they implemented the ability for users to change the ordering of rules. `iptables` does not support reordering of rules either. Given this, it can be concluded that forming rulesets specific to your network with proper ordering can be challenging. As for Cloudflare, users would prefer the feature to reorder so that they can update firewall rules easily according to existing policies.

2.8 Summary

Firewalls are a robust security system to prevent unauthorised users from accessing computing resources on a private network [13]. There are many different types of firewalls, each having their purpose. `iptables` is one way to configure firewalls in Linux, providing a command line to set the rules of different chains in the firewall.

Firewalls do however have their limitations. A firewall can not detect an attack originating from within the network of computers that the firewall protects. Ordering the rules of a firewall can also be challenging when configuring a firewall with a large set of rules, as some rules might shadow the rule you wish to set. Given that many users requested Cloudflare to allow the option to reorder rules instead of deleting and adding them manually, it can be concluded that easier ways to order rules are a highly sought feature for firewalls.

Chapter 3

Educational Games

In this chapter, we will first discuss different ways games can help users learn new information. We will also discuss different ways games can engage different users. We will then analyse different games for teaching security concepts to a user; what makes these games good and what we can learn from them.

3.1 Games for education

Learning a task that requires mental engagement needs a lot of effort from students as well as a lot of motivation to cope with the task [14]. Alice Mitchell and Carol Savill-Smith suggested a few reasons video games can engage users in learning new information: [15]

- Games motivate fun learning
- Complex games provide interactive, virtual environments which further engage users
- Including sound in a game creates a more immersive experience that could keep the interest of the user in the game.

Due to this, video games have been used to teach many different topics. There has been a recent development in video games to teach different aspects of security.
Player engagement in a game is important as it determines if the player would continue to play a game or not. There are four identified types of engaged-behaviours [16]:

- **Environment Directed:** These are players who are interested in the world environment of the game. Environmental engagement depends on the vividness of the immersion, on the variety and aesthetics of the world, and on the possibility of exploring or modifying the environment.
- **Social-Directed:** These players enjoy the social connections within a game, i.e. players who want to expand their social relations to other players. These social connections can also be formed with virtual characters within a game. If there

is a virtual character, it is important that they do not display stereotyped speech and behaviour as players like to perceive some intelligence from the character.

- **Self-directed:** These are players that like to customise a personal avatar within the game. They like to spend time studying features and accessories within a game. These players also like role-play.
- **Action directed:** These are those players who are interested in the main core and challenge of the game. They try to finish challenges quickly and might even challenge themselves personally, e.g. time how long it takes them to get through a level.

Different games target different types of player engagements, some aiming at a subset of these types while some target all of them. These types are interesting as they provide a basis for the different elements a game could have to appeal to different kinds of users. In my first game, I decided to target action directed and environment directed players. We will now discuss different security games and the different ways they engage different types of users.

3.2 Security Games

3.2.1 Zachtronics

Zachtronics [17] is an independent video game development studio that design engineering puzzle video games which include a variety of different security games. One of their games is Exapunks [18], which is a game to teach its players 'opcodes', a machine instruction language used to the operate a system [19].

This game received a positive reception on Steam, which signifies that the game is quite popular [18]. What makes it security based is the fact that the player is a hacker who uses these 'opcodes' to break into computer systems to afford a drug.

Another great game by them is TIS-100 [20], in which the player has to develop assembly language code to perform certain tasks on a simulated computer that is corrupted. The game mostly had positive reviews on Steam [21].

The reviews for Exapunks [22] and TIS-100 [21] are overwhelmingly positive as written on the Steam website. A majority of the good reviews are due to the game being hard, but not overwhelmingly difficult, thus still presenting a challenge to players. These reviews also suggested that the nostalgic elements of old technology enhanced the game experience and give the impression that challenges can motivate players to continue to play a game, which represents action directed players that we discussed in Section 3.1.

From this, we can understand the level of difficulty the challenges in our game should have to keep players engaged within the game.

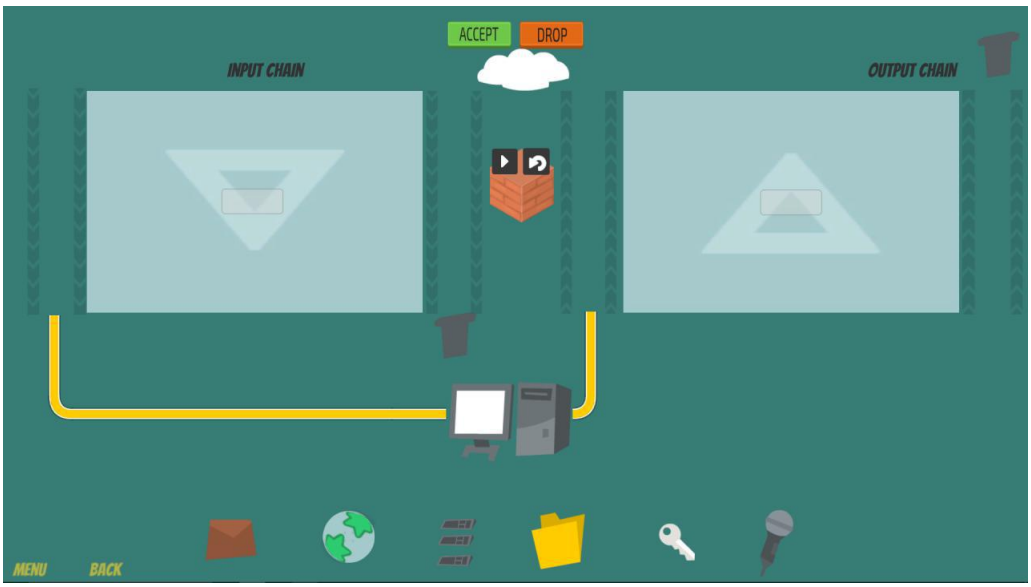


Figure 3.1: Permission Impossible Level Layout [23]

3.2.2 Permission Impossible

Sehl et al. created a game called Permission Impossible which aimed to teach the concept of chains to average university students [23]. This game allowed its players to drag and drop different rules to the different chains to simulate setting up a firewall, as shown in Figure 3.1. The purpose of this game was to answer three main questions in the dissertation [23]:

1. Is a video game suitable to teach a target group of beginners and novices about firewalls?
2. Has the target group increased their knowledge and learning according to the learning goals identified?
3. Can the target group identify firewall terminology and concepts?

After the evaluation of the user studies of this game, Seht et al. mentions different ways to improve the game. She says that the game can be made to be more visually appealing, such as using cables to connect computers to make it clear to players. In addition to this, she mentions that rewarding the user for completing the game, e.g. a congratulation message provides more motivation [23].

These improvements are something we can consider in our own game to enhance the player experience.

3.2.3 Blue Team

Blue team is a game that aimed to teach players “the logic behind setting up firewall rules, some of the basic syntax used, and show examples of what the logs produced by

IDS may look like when under attack." [24]. This game had many interesting elements such as an interactive network diagram, which allowed players to click on a computer to view the services running on it, and on the firewall to configure it. It assumed some prior knowledge of rules set up, as it essentially does not teach this during the game. Its target audience were people who had already taken the computer security course and wanted to enhance their firewall knowledge.

While the game has no proper feedback or user study, I found it an engaging user interface that clearly outlined the network and provided all the necessary information to solve it. The 'hint' feature in the game was advantageous and helped me get through the levels without revealing the answer. From my personal experience, I can say the best elements of this game were the network diagram, as it provides a good view of how the network is set up, thus helping me understand how to play the game with minimal text. Due to this, I planned to incorporate elements of this game into my own, specifically those I found pivotal to engagement.

3.3 Summary

In this chapter, we discussed why video games might be effective in teaching players about different topics. We then analysed different security educational games using either their user studies, user reviews or just a personal review about what they did to engage players into their game. We also discussed points that the creator of some of the games felt might have engaged the players further into the game.

Chapter 4

Game Design

This chapter describes the different technologies and tools used for the creation of this project. We go through an initial design for the game and explain why we chose to design the game in this way. Upon realising the limitations of the initial design, we discuss an alternate approach to remove the restrictions that the initial design provided.

4.1 Technologies Used

4.1.1 Programming Language and IDE

Using an Integrated Development Environment, or IDE, is hugely beneficial for programming, as it provides useful features such as code completion and error checking. These features can significantly improve the speed and quality of code written for any project.

After considering different IDEs for game design, We chose Unity for this project. Unity is a game engine - an IDE for video game development - that provides many useful features. It has a convenient drag and drop interface for different elements within a game. This interface reduces the need for implementing many features such as animations, as Unity provides this tool within its IDE. In addition to this, Unity supports many platforms including Windows, Linux, Android and WebGL. WebGL is a graphics API that is used to render graphics within a web browser. The game has been designed to compile into WebGL format. WebGL was chosen to allow the game to be played online regardless of the user's operating system. While Unity does support many operating systems, optimising it for different screen resolutions and operating systems would not be trivial. The option to build the game into WebGL format allowed it to automatically adjust to different resolutions and run for all operating systems that can access a web browser.

Unity supports C# for its game design. Due to my prior experience in Java, C# was a good option as the two languages are quite similar in programming syntax. While

the API of Unity in C# took time to get used to, it was a good experience and reduced immense amounts of redundant code such as programming animations, which would otherwise have to be implemented from scratch.

While Unity is generally a good option for game design, there were many unneeded features such as the physics engine in the game. Though the final build of the game does not include any un-used libraries. In addition to this, some of the User Interface elements such as the creating a table were quite tricky to implement and would perhaps have been simpler to create using HTML, CSS and JavaScript, as the game is compiled to a WebGL format.

4.1.2 Graphical Design

The art for the user interface and elements present in the game are created using Adobe Photoshop [25]. I had prior experience in using this tool. In addition, it provides many useful features that allow precise pixel placement for different elements of a game. Due to it's capabilities to render different sizes and compress images to smaller file sizes, it was the optimal choice. Hence it was chosen to design the user interface elements as well as any art used in the game.

Blender [26] is an application that provides different tools to create 3D models. Blender has been used to create 3D elements within this game to provide a more aesthetic appeal to the game.

4.1.3 Version Control

Git [27] is a version control system that provides the capability to track changes in source code during the development of software. Having used Git for many projects in the past, it was an optimal choice. Git also provides the option to go back to a previous version of a project if you wish to, which is highly useful if an error occurs with the present version of the software.

Github [28] is an online website that allows version control and backup of a project using Git. It provides an online backup for the code in case the local system used for the development of the project fails. The project is private in GitHub - the public can not view it. I do intend to make the project open source in the future, i.e. publicly available.

4.1.4 Other Game Assets

The music and the art used in this project is available publicly on OpenGameArt [29]. This website allows people to download and use people's assets publicly available on their site. While some assets require crediting specific people, the assets used in the

game do not need special credits. Open source game assets were used to avoid recreating aspects that were publicly available. This saves time and helps make development of the game more efficient.

4.2 First Design

4.2.1 Concept

My initial goal was to teach the simple commands and syntax of `iptables`, which includes basic commands of adding rules to a firewall and understanding the use of chains explained in Section 2.6. I also intended to teach players about the Transmission Control Protocol, or TCP for short, as well as the different application protocols that use it, including HTTP and SSH.

My target users for this game were university students who wanted to learn `iptables`, irrespective of which school within the university they belonged to. As mentioned in Section 3.2.2, 'Permission Impossible' [23] gave future work suggestions mentioned which were intended to be in the game but could not be. These suggestions included a screen to congratulate the player for playing the game and making sure the concepts were more apparent to a user. To ensure that concepts taught in the game were easier to understand, I decided to visualise packet movement in this game, showing a high-level view of how a firewall controls which packets go through which ports. This visualisation aimed to help players understand how packets move within a network at a glance. It was also intended to visualise other concepts within the game to aid explanations.

4.2.2 Intentions

In the first iteration of the game, we decided to create an engaging story which would catch the interest of players. Furthermore, the game intended to appeal to two of the four types of players engagement behaviours discussed in Section 3.1, namely the environment directed and action directed. To engage environment directed players, the game was designed with a compelling story. To engage action directed players, another focus of this game was to make it challenging to players.

4.2.3 Story Design

My story was as follows:

"Facebook, Amazon, Google and Twitter were ruling the world, but they realized they still had one big challenge to overcome, each other. Determined to win, they each built a time blasting device to force the other company back into the past in a time where electronics did not exist. Unfortunately, they did not realize that they were all firing

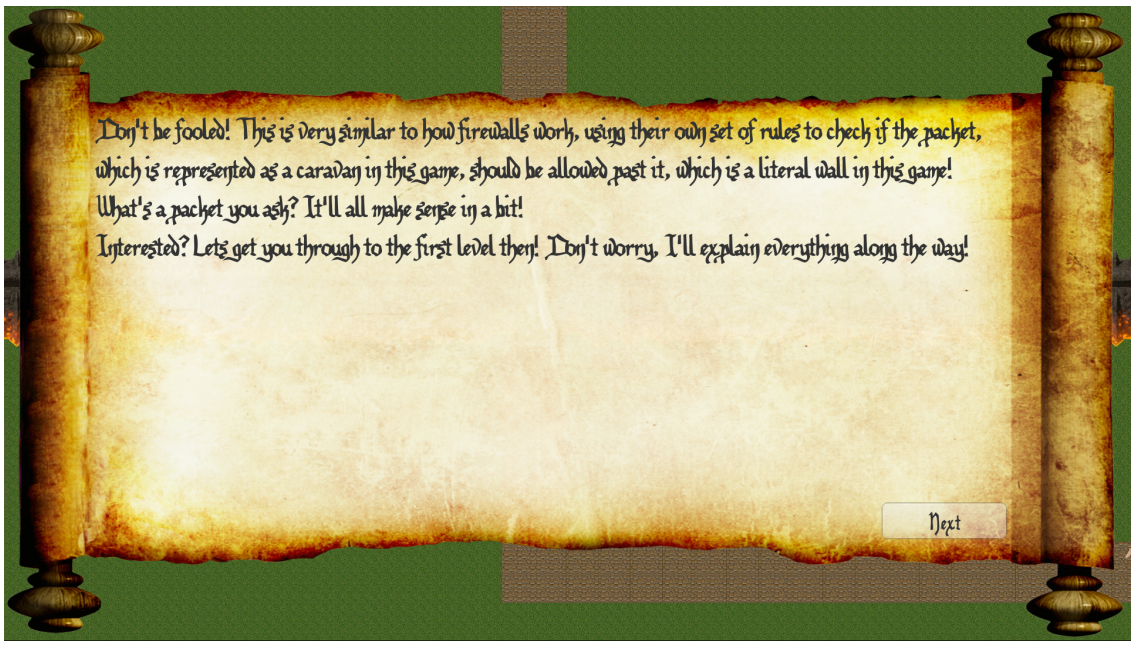


Figure 4.1: First Game Instructions Scene

these devices at one another, forcing all of them back to the past together! Now that they had been removed from their original timeline, they thought to do the only thing they knew best, Rule the world! They set up their own cities. Given that they were all technology people, all they knew were technology concepts. So they all decided to operate their trading system based on firewalls! Using security guards at each gate, they controlled which caravans were allowed into and out of their cities, similar to how packets are either allowed or denied into a network."

This story possesses nostalgic elements to real companies which might be relatable to some players inspired by the nostalgic elements in the games created by Zachtronics. [17].

4.2.4 User Interface Design

The graphics and layout of the game were made in an old style to capture the aesthetic appeal of historical times.

The instructions screen, as shown in Figure 4.1, is used to convey the game's message to the player, which includes the story and any concepts the game wishes to teach. The text was carefully planned to keep the player engaged through it's comedic tone. The first level of the game can be seen in Figure 4.2. Most levels would be similar in design to this with minimal changes depending on what the level requires, e.g. instead of the Facebook building it could be an Amazon building. The caravan, as shown by label 1 in Figure 4.2 and represents a packet that was sent by Twitter in this level to Facebook.

The gates, as shown by label 2 in Figure 4.2, represent a visualisation of ports in a

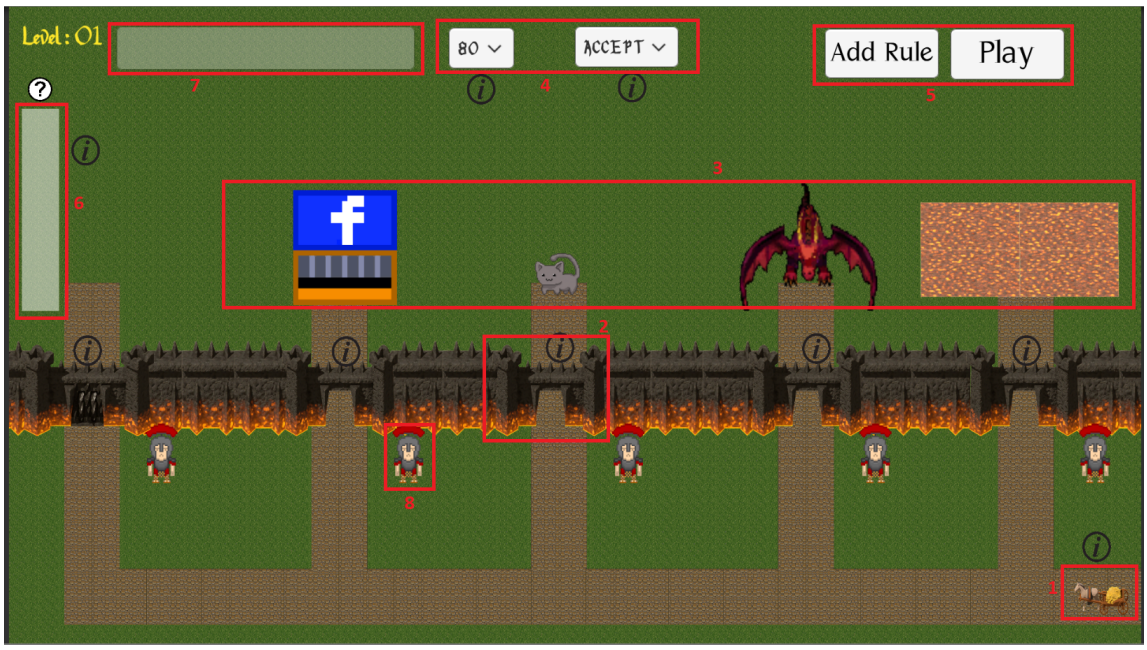


Figure 4.2: First game level layout, the caravan labelled as 1, a gate labelled as 2, the characters labelled as 3, the drop-down menus labelled as 4, the buttons labelled as 5, the rule number panel labelled as 6 and rule description panel labelled as 7

computer. The guards, one of them labelled as 8, together form the firewall, obeying the rules set by the player as a normal firewall would. This level intended for the player to configure the firewall to allow the caravan to pass through the gate in front of the Facebook building (the left most element in box labelled 3). The player would use the drop-down menus, labelled by 4, to select options for the rule. In this case, the first level only introduced ports and whether to ACCEPT or DROP a packet, i.e. Allow the packet or block it. Each element had an icon near it which if hovered upon would describe the user interface element to the player. For example, hovering over the caravan would show it was from Twitter in this level, hovering over different gate's would show their port number e.t.c.

The player could add a rule by pressing the 'Add Rule' button (labelled as 5 in Figure 4.2). The panel labelled as 6 in the Figure would show the rules listed by only their rule number i.e. the first rule added by the player would be show by the number 1, the second rule would be shown by the number 2 e.t.c. When a player hovered over this number, the actual rule would be displayed in the panel labelled as 7. The player also had the option to remove a rule by right clicking on a number in the panel labelled as 6. When the player pressed the 'Play' button as shown by label 5, the caravan would begin to move, and depending on the rules the player had configured, go to the Facebook building. It would then win the game and proceed to another instructions scene for the next level.

To enhance the game aspect, behind each gate was a different character (all labelled as 3 in Figure 4.2) to warn the player not to let the Caravan go through that port e.g. If the Caravan went through the gate in front of the 'dragon', a warning screen would show

up informing the player that the caravan was 'eaten by the dragon' and asked to try again. The caravan would go to each gate and only go through if the player configured the firewall to allow it 'ACCEPT' it through the specific gate.

This implementation of the game did have its drawbacks. The level included many different elements which would require more time from a player to accustom to the level design. While there was a UI tutorial in the game, it was not enough for a player to understand the layout. This is further addressed in the next section.

4.2.5 Game Analysis

To assess the usability of the system, two participants were tasked to navigate through the game and play through the first level. This testing was not a formal study; however, it was vital in understanding what difficulties players would have with the interface of the game. Given the many different elements in the level, it was expected for players to be overwhelmed by the game screen. To avoid this, the user interface tutorial was added. However, despite this tutorial, both users faced the same issue of not understanding what they had to do without guidance.

Given that the game already attempted to explain the user interface to the player, it was not enough to cover every element present in the level. This suggests that having fewer elements within the game interface might be beneficial to players. Given this was an educational game, learning to use the interface should be trivial as the challenge of the game should be the actual information taught by the game.

4.2.6 Learning Outcomes

The target audience of "university students who wanted to learn about `iptables`" was too broad. For instance, I failed to consider that different people have mixed experience with computers, and would hence require different ways of explaining certain concepts. If the explanation was too simple, an informatics student in their third or fourth year might not find the game very engaging, and if it was too complicated, a informatics student in their first or second year might be unable to play it. Due to these reasons, I believed the game required a redesign before I started proper user studies.

Designing this game did have its benefits. This game allowed me to become more familiar with the Unity IDE, hence allowing me to start the redesign with a clear image of how to do things. Having been fortunate for having faced issues like this early on, we were able to learn and improve on this within a reasonable amount of time.

The learning outcomes of the game were also not justified properly. There was a plan to detail what each level would teach the player, however, there was no reason as to why the game would teach those specific aspects about firewalls. Due to this, we

decided to gather proper requirements from our target audience to understand what is important for that specific target audience to know about firewalls.

4.3 Second Design

This section will describe the concept for the new iteration of the game. We will set our target group for the game. We will then detail how we gathered the requirements of the game by surveying the target group. From these requirements, we will then list the learning goals for the game.

4.3.1 Concept

Given my rather broad target group written in Section 4.2.1, narrowing it down was essential. I decided to target informatics students only, as they were the most available to me personally, which would assist in finding participants for user studies.

From my personal experience of learning `iptables`, I remembered it to be a challenging part of the Computer Security course (A course taught to 3rd or 4th-year Informatics students at the University of Edinburgh [30]). While there were a few games that taught `iptables`, they did not properly target the concepts taught by the course. This is why I decided to create a game to assist the learning of `iptables` and firewalls in the Computer Security course.

4.3.2 Requirements

4.3.2.1 Gathering

Firstly, I went on to confirm if other people would have preferred a game environment for the learning of `iptables` in the Computer Security course. I also wished to understand what people found hard in the section of the course that taught about firewalls and required `iptables` knowledge. In the Computer Security course in the year 2017-2018, there were three courseworks in total. In coursework one, there was a section in which the students were tasked with configuring firewall rules using `iptables` by following instructions on what traffic to allow and deny.

My method of gathering the information from the users was through an online form hosted on Google Forms [31]. This questionnaire was created to query students with certain questions regarding the coursework, which is attached in the Appendix B

This form gathered the feedback students got for the questions that required `iptables` knowledge. Through this, I gathered the feedback they received from the coursework marker about what parts they got wrong as well as the students feedback on how they regarded the `iptables` part of the course.

What, if anything, did you find hard about those questions?

5 responses

There was a lot of ambiguity and guessing what is right and what is wrong from what I remember
The commands
Having little experience of iptables before beginning the coursework.
Syntax
not hard

Figure 4.3: Responses when asked what participants found hard in Google Forms Questionnaire

The online questionnaire had five responses which were not adequate to confirm difficulties that the majority of the students might have faced. Thus, I asked people who did not wish to fill in the questionnaire what they found hard.

4.3.2.2 Results

While the online questionnaire only had five responses, it provided feedback in terms of what marks they lost, and what they felt could have helped them in the `iptables` section of the course. Figure 4.3 shows a response to the question asking students what they found hard about questions related to `iptables`. From this, we can conclude that students generally found the commands a bit overwhelming. They also found it tough to understand what commands to put in and what should be blocked or allowed. One student also mentioned in the forms that lack of experience in `iptables` before doing the coursework led to them finding it difficult.

Figure 4.4 shows the response of students when asked what other resources they used to assist them in learning about `iptables`. Googling general commands and checking out a guide regarding `iptables` online seem to be the most common answer. Only one response is from someone who has played a firewall game in the past; however, they did prefer to have a game that specifically teaches them the syntax rather than read the documentation of `iptables` to understand them. In addition to this, three people were asked different questions through direct messaging. When asked what they found hard, they said that they struggled to get the right rule on the right chain. In particular, one person said that understanding the syntax of `iptables` was something they found tricky. When asked what specifically they found hard about the syntax, they said they struggled with including all of the required information in one command.

Given this information, we can extracted a set of requirements the game must fulfil in order to assist students in this course:

What, if any, further resources did you use to learn about 'iptables'?

5 responses

Googling
Cisco networking guide
I used some given Firewall games. I thought these were very useful but it would have been even better to play a game teaching me the syntax of iptable commands rather than pouring over a bunch of documentation to learn about this.
Google
youtube

Figure 4.4: Responses when asked what additional resources were used for the course in Google Forms Questionnaire

- **Requirement 1 :** The game must teach the syntax of `iptables` in a user friendly manner.
- **Requirement 2 :** The game must ensure that players understand what rules belong in what chains
- **Requirement 3 :** The game must ensure that the protocols associated with firewalls should be clear to make sure that they know what they are permitting and what they are blocking.
- **Requirement 4 :** The game should have aspects appealing to some of the engagement behaviours presented in Section 3.1.
- **Requirement 5 :** The game should build upon future work suggestions from past games designed to teach firewalls such as those listed in Section 3.2.2.

4.3.3 User Learning Goals

As an overview, the learning goals derived from the user feedback are as follows :

- The syntax of `iptables`
- The INPUT and OUTPUT chains
- TCP and UDP, the basics enough to understand firewall rules
- The application protocols HTTP and SSH, just the basics to understand firewall rules
- The different ports regarding the different application protocols.

Thees learning goals do not include everything that the Computer Security course teaches about `iptables` such as connection states. These were restricted so that the

game can focus on key concepts that could be expanded on in the future after evaluating the game with the subset of concepts first.

4.4 Summary

The game for this project has been designed in Unity and programmed in C#. We designed and implemented an initial game which focused on aesthetics to enhance player engagement. After having two students play this game, there was a problem concerning the usability aspect of the game. We then narrowed down our audience to informatics students who had not taken a course on Computer Security. After surveying students who had taken the course, we gathered requirements to target particular issues students faced with `iptables` in the course. Using these requirements, we derived a set of learning goals for the new game.

These goals were narrowed down to cover a subset of topics that are taught by the computer security course regarding `iptables`. These could be expanded on in the future after evaluating the game with the target audience.

Chapter 5

Game Implementation

5.1 Iterative Development

This game has followed an iterative development approach, where it has been changed based on feedback from evaluation. In this chapter, we present and discuss in detail the final iteration of the game. We detail the game's level design and layout and how each level intends to meet the different learning goals mentioned in Section 4.3.3. Each level in the game builds upon knowledge from preceding levels to ensure the player does not forget information that is taught before the current level they are on. We will then detail each level's design and learning goals. After further analysis of the games resource usage in a system, we will derive system requirements to run the game.

5.2 Game layout

For the new game, a simple layout with only necessary elements was planned to minimize user confusion while playing the game. To teach the player about different information and to instruct them on their tasks, a scene where the player was told information and objectives for the level was made, which is called the briefing scene.

To deliver the briefing to the player, I decided to create a character with a unique personality. This character targets players that have social-directed engagement behaviours, mentioned in Section 3.1. The reason this targets such engagement behaviour is due to the virtual character's personality within the game. The general personality planned for this character was a superiority complex, i.e. the character starts by seeing the player as a novice in respect to firewalls and `iptables`. As the game progresses, this character's personality starts to become more friendly as he starts to realise that the player is becoming smarter which eventually causes the character to accept the player as a valuable person. My choice for this character was more focused on a comedic aspect, and thus, I decided to make this character a cat, and named this character 'Colonel Whiskers'.

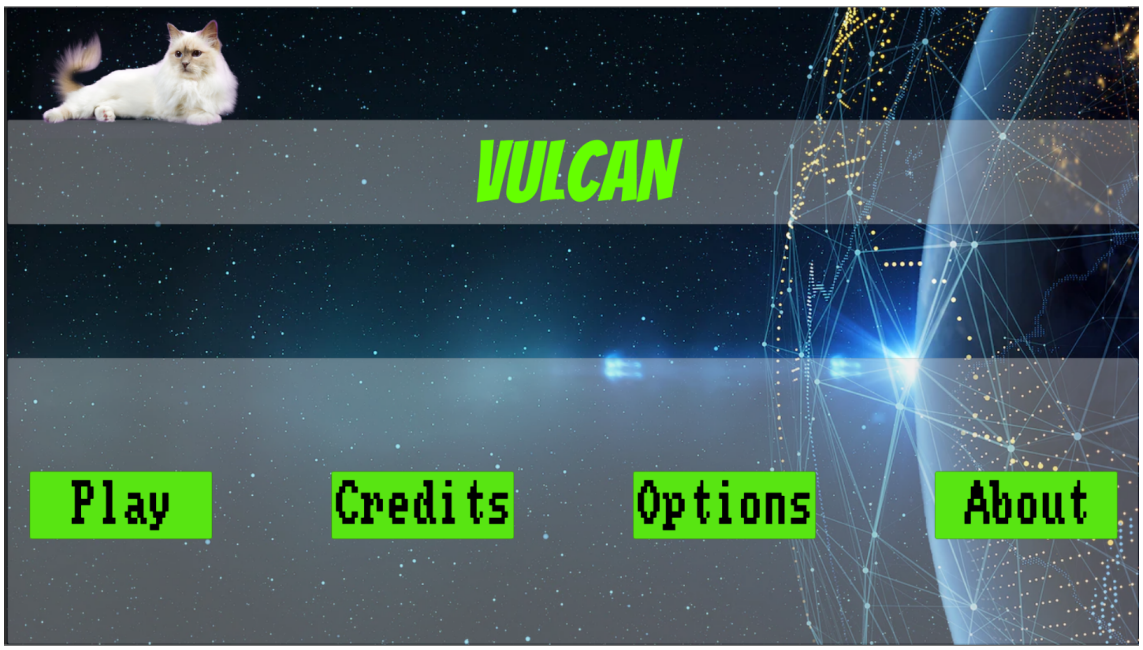


Figure 5.1: Main Menu Screen of the Game

Given this is an educational game, my main focus within this game, other than social-directed players, was to target action-directed Players mentioned in Section 3.1 i.e. players who's engagement is due to the challenges of the game. The game presented a new challenge at each level, building upon the current knowledge they had attained from the briefing before the level. Some of the later levels presented in the game can be solved in multiple ways, which could appeal to action-directed players to figure out all the possible ways they could solve a level.

5.3 Game Scenes

The game starts with the Main Menu screen, shown in Figure 5.1. There are four buttons present on this screen, which are the **Play** Button, the **Credits** button, the **Options** button and the **About** button. The scenes that loads for the **Options**, **Credits** and **About** button can be found in the Appendix A.

When the user presses the **Play** button, the game begins, loading the **Briefing** Scene. This scene is shown in Figure 5.2. It is composed of 'Colonel Whiskers' - the character who gives the briefing, the explanation and the objectives to the player, a briefing panel which displays the text to read, and two buttons to go back and forth in case the player wants to view some information again. It also consists of a counter at the bottom to let the player know how many more briefing pages they have to read.

The font has been chosen to maximise readability for the player. The command is highlighted with a different colour to be noticeable by the player. I assume that as

Figure 5.2: One of the Briefing Screens of the game

the target users are from the school of informatics, they understand what a command means. After this, the player goes through a tutorial that informs the user how to navigate around the level. Figure 5.3 shows a section of this tutorial. The **INPUT** and **OUTPUT** tabs on the top right indicate the different chains that `iptables` has. Clicking on them will open that particular chain. There is a terminal in which the player has to write out commands to configure the firewall.

The firewall itself consists of 6 columns to assist the player in understanding and confirming the rules they set. There is also another column to show the line number of the command when the user adds it to the chain. An example of the screen when a rule is added is shown in Figure 5.4. There is also a default policy shown for each chain, written on the table as "Default: ACCEPT" which means that the default policy is currently **ACCEPT**.

Below the firewall interface is the terminal. The terminal has been made using an input field where the player can type in commands for `iptables`. The commands are parsed, and different actions are taken depending on what the command is, e.g. if the command is to add a rule to the **INPUT** chain, the UI element for the **INPUT** chain shall be appended to contain the new rule as well. The command can be executed by either pressing the "Run" button or pressing the "Enter" key on the keyboard.

The game also has music that was selected based on personal preference on how engaging it made the game. In Section 3.1, we discussed that one of the ways to engage users in learning new information is to include sound in a game. This is to create an immersive experience to maintain a player's interest within the game. This is why music was chosen for the game.

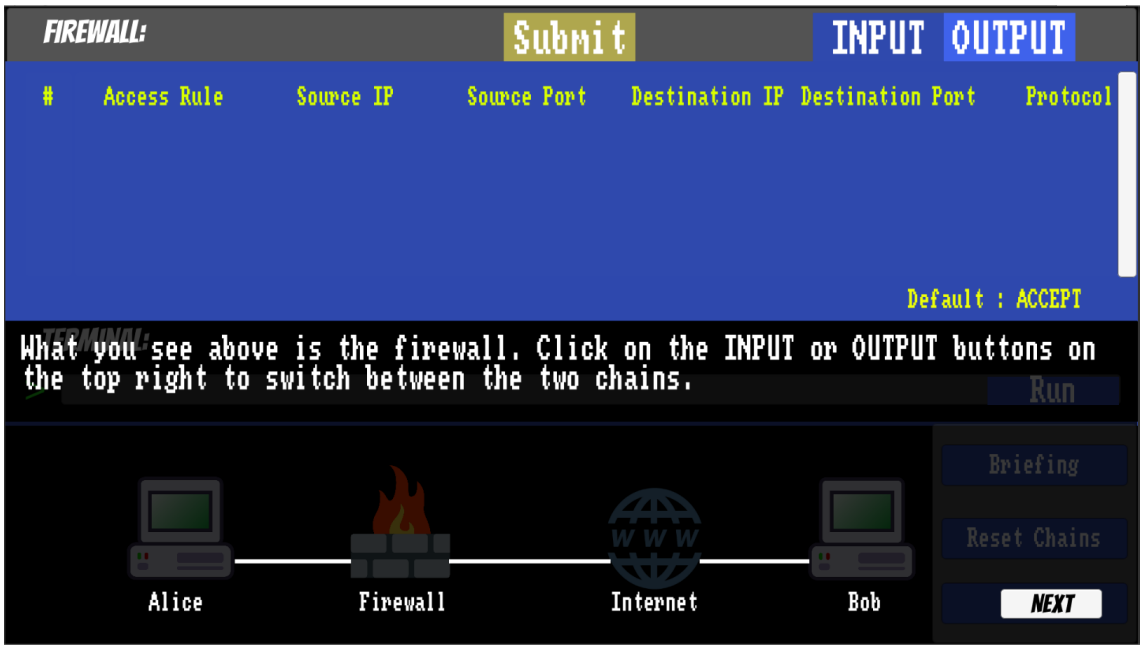


Figure 5.3: First level UI tutorial of the game

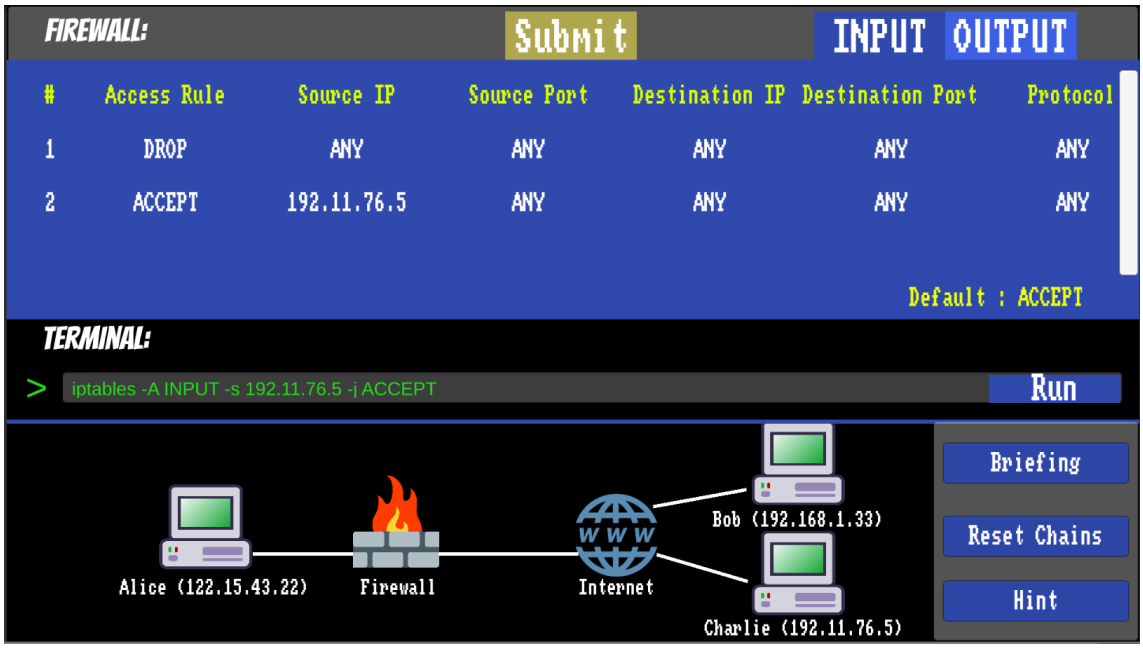


Figure 5.4: A level after rules are added

We will now go into detail about each level, which includes what is required to win the level as well as what each level wants the player to learn on its completion.

5.3.1 Level Design

The levels have been designed to build upon previous knowledge from possible preceding levels as well as test new knowledge from the player.

5.3.1.1 Level 1

The first level aims to introduce plenty of new information to the player such as the concept of a firewall and `iptables`. It also explains what the **INPUT** and **OUTPUT** chains mean. The last thing it tells the player is what a default policy is.

The challenge this level presents to the player is to set the default policy for both **INPUT** and **OUTPUT** chains to **DROP**. The level starts with both chains having a default policy of **ACCEPT**. The command to change the default policy is taught in the briefing by Colonel Whiskers. In this level, the player can view the command again by pressing the briefing panel. The hint for this level reminds the player to set the default policy for both the **INPUT** and **OUTPUT** chains.

5.3.1.2 Level 2

The second level aims to teach the player how to add a simple rule to any chain. It also introduces the concept of IP addresses, and how they are used by firewalls to block or allow traffic. It evaluates this by asking the player to allow Charlie's traffic into Alice through the firewall, and **DROP** everything else. The player can view the IP address or Charlie in the network diagram below the terminal, as shown in Figure 5.4.

The briefing page in this level reminds the player of the command to add a rule. In case the user forgets the command to set the default policy, it is in the hint panel. The player is not reminded about the command to set a default policy in the briefing after this level, as one of the aims of the game requires them to learn the syntax and commands.

5.3.1.3 Level 3

The third level introduces the Transmission Control Protocol (TCP) to the player. It tells the player how it works at a high level, i.e. how it maintains no packet loss. The level requires the player to allow only TCP traffic through the firewall specifically, and **DROP** everything else. While this might confuse some players to believe that this might allow all kinds of traffic if they are under the assumption that everything uses

TCP, this is also to introduce the notion that there are other protocols in network communication.

When the player wins the level, a UDP packet comes out of Charlie's computer and collides with the firewall, where it does not go through. While the player has not been introduced to UDP yet, this is to show that there are other types of protocols that packets use which the current rule set won't allow through the firewall. The player is taught about UDP in later levels.

The briefing panel of the game tells the player to allow Charlie to SSH into Alice's computer and to drop everything else. It also suggests using the default policy for the second part. The briefing panel does not have the syntax for the default policy, but it is included in the hint if some players still do not remember the syntax.

5.3.1.4 Level 4

HTTP is introduced in the fourth level. The player is instructed on what it stands for as well as what it is used for. They are also told that HTTP commonly uses TCP, and for this game, they should specify HTTP in the `--dport` parameter with TCP as the `-p` parameter in the command. To win this level, the player has to block all HTTP traffic into and out of the firewall. As this level doesn't specify any traffic to be allowed, the player can solve this level in multiple ways, which the game allows.

The briefing reminds the player of the command in case they have forgotten, as I believe it is still an early stage for them to remember the entire command without making some errors.

5.3.1.5 Level 5

The fifth level tells the player about the SSH protocol. It tells the player what it stands for and what it is used for. Colonel Whiskers starts to get more impressed by the player at this point and changes his attitude to be more positive towards the player. It also tells the player that SSH requires the firewall to allow communication both ways from one computer to another. It evaluates this information in the level by requiring the player to allow Charlie to SSH into Alice's computer, and **DROP** everything else, which requires setting the default policy to **DROP**.

An example for the command is mentioned in the briefing panel, as it will probably still take time for a player to remember the command.

5.3.1.6 Level 6

The sixth level introduces the User Datagram Protocol. It tells the player what UDP is, and in which situations it used over the TCP protocol. The level requires the player

to block HTTP traffic from Alice to Bob, and from Bob to Alice. Also, the player has to allow Charlie to send UDP packets to Alice. There are multiple ways to solve this level. HTTP and TCP are also included this level to start letting the player experience different rules concerning different protocols.

While the Colonel does give an example of a command using UDP, the briefing panel in the level does not show the player the command. The reason for not mentioning it is to allow players who want to remember the syntax to test themselves. In case the player really can't remember the command, the hint panel contains an example of the command.

5.3.1.7 Level 7

The seventh level is meant to be the hardest level of the game. It tells the player why ordering of rules matter and how a packet is matched according to its rule. It requires the player to do four things: not to allow Alice to SSH into Bob's Computer, allow TCP traffic from Charlie's computer through the firewall, allow TCP Traffic from Alice's computer and to set the default policy of both chains to **DROP**. The player is told that in this level, the order they do this in matters in this level. Setting the default policy is independent of order. This has been added to see if the player remembers that the default policy differs during the think-aloud, althought it is never directly assessed.

The briefing panel shows four tasks that the player has to do by numbering them. In this level, the player is not reminded of the command at all, to assess if the player has learned the commands for doing this. The hint reminds the player that SSH requires both chains to allow it through the firewall.

5.3.1.8 Level 8

This level teaches the player how to delete a rule by its corresponding line number. The player's task is to delete a rule so that Alice can send packets to Charlie.

The briefing panel displays the user's task as well as the command for the level. The hint is that the player just needs to delete one rule to win.

This level is the third level in the final iteration of the game due to reasons discussed in Chapter 7. After this level is set as the third level, all other levels numbers are pushed i.e. the original third level now becomes the fourth level, the original fourth level now becomes the fifth level e.t.c.

5.3.1.9 Winning Screen

Once the player wins the game, they are presented with a winning screen. The image for the winning screen is attach in Appendix A.4 This consists of a congratulation

message along with a fun video showing kittens which appealed to players. This video was taken from Youtube and is allowed to be used by anyone if they so wish to. This video was chosen to maintain the theme of cats within the game. Based on evaluation discussed in Chapter 7, this was considered to be a good winning screen and is referred to as 'the prize' of winning the game.

5.4 Game Statistics

5.4.1 Lines of Code

This project consists of multiple scripts for each interactive element in the game screen. Using the Powershell command line in Windows 10, as shown in Figure 5.5. I counted the lines of code written. The lines of code for this project are 8889. This includes lines that show package imports, thus, it would probably be less than this.

```
PS C:\Users\Rusab\Documents\Vulcan - Firewall Administration\Assets> dir . -filter "*.cs" -Recurse -name | foreach{(GC $_).Count} | measure-object -sum

Count    : 71
Average  :
Sum      : 8889
Maximum  :
Minimum  :
Property :
```

Figure 5.5: Powershell script to count Lines of code

5.4.2 Build Size

After building the game to WebGL format, the size of the game is 65.6 Mb, as shown in Figure 5.6. This big size is due to the high quality of images used in this game. Within this folder, the game is run by clicking the 'index.html' file. The game will be uploaded online in the future, thus not requiring any local download by any user.

5.4.3 Resolution

The game is build to a resolution of 1920x1080 by default. Currently, there is no option to change this resolution, as the game is optimised to that specific resolution. However, the game does try to adjust to a player's resolution by inserting black bars to adjust the game size.

5.4.4 Memory Usage

To analyse the memory usage of the game, the Unity memory profiler was used. This showed a breakdown of the usage throughout the runtime of the game [32]. This profiler can be used to analyse the memory usage for a game within the Unity editor. Figure 5.7 shows the game's memory usage. To capture this memory usage, I played the

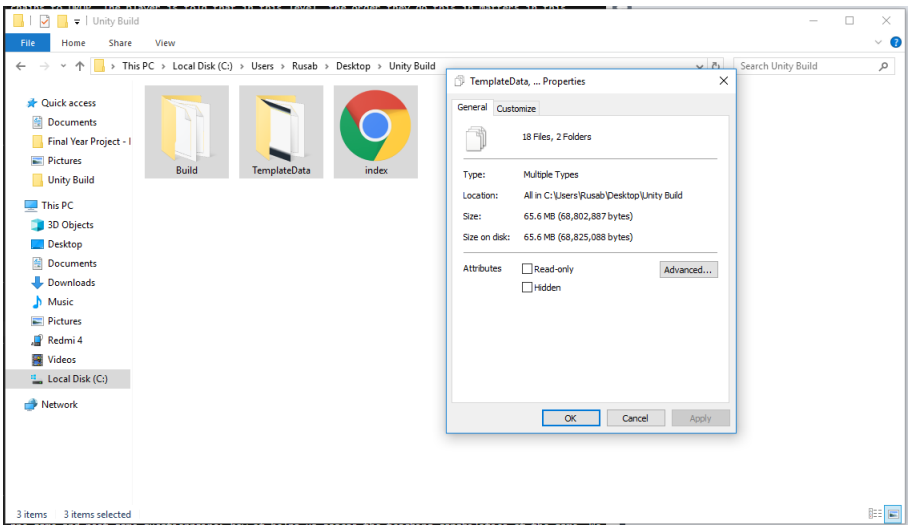


Figure 5.6: The folder containing the game's build into WebGL

game from the start of the main menu until I completed the first level. The total system memory used was 1.03 Gb, which is the total memory used by the game throughout its execution. This memory usage also includes Unity's allocation for memory to run the game. Thus the total system memory usage for the actual game would be less than this.

Unity reserves memory pools to avoid having to ask the operating system for memory too often, which is shown by the reserved total parameter being 448.7 Mb. At the end of the finishing the first level, the memory currently being used at that point was 234.6 Mb, which is shown by the used total parameter.

The profiler shows the memory usage only when running the game within Unity. To analyse the memory usage of the game in a browser, we observed the change in RAM usage of the Firefox browser before and after running the game. Figure 5.9 shows the difference, which results in the game taking 447.8 Mb of RAM.

5.4.5 CPU Usage

The Unity profile has again been used to analyse the CPU usage, as shown in Figure 5.8. The profiler provides the option to focus on any particular point of execution of the game; in this case, we focused on the process that took the longest time to finish. Upon further investigation, the 'Gfx.ProcessCommands' takes a total of 925.97 ms to be done. This process renders the graphics within the Unity engine and thus would take the longest the instantiate. Also, this process starts whenever a new scene has to be loaded, i.e. when the player goes from the briefing scene to the first level scene. Due to Unity loading a new scene and hence new graphics, this process has taken more time than normal processes, which take less than 50 ms.

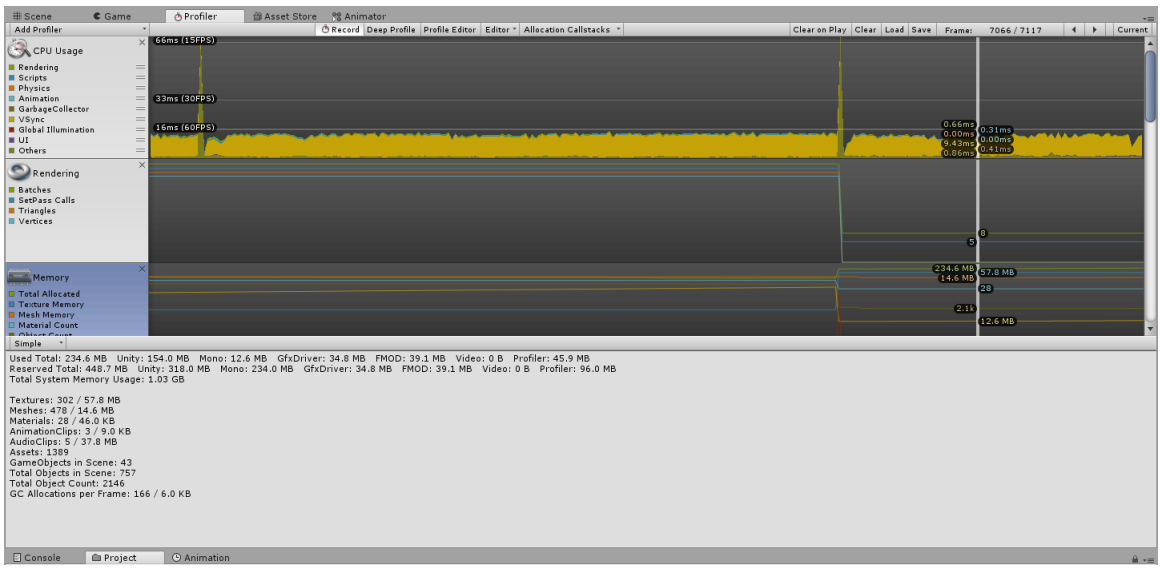


Figure 5.7: Unity Profiler showing memory usage

5.5 System Requirements

Given that this game is built to WebGL format, the requirements to run the game are within the browser's capabilities to render graphics as well as manage memory. However, upon the analysis mentioned in the prior sections, we can conclude that running the game requires at least 1.5 Gb of RAM, as the highest RAM usage by the game was 1.03 GB throughout its execution.

In terms of CPU, we cannot derive stringent requirements, as our analysis only focused on what processes within the game took the longest to complete. Having tested the game on an Intel Core i5-6200 CPU [33], the game ran smoothly and efficiently with no drop of frame rate.

5.6 Summary

The game consists of eight levels, each level teaching the player a new command or topic on firewalls and protocols. The interface includes the actual firewall which shows both the **INPUT** and **OUTPUT** chain with the rules listed for it. It also consists of a terminal for the player to type in the commands to configure the firewall. Below this is a network diagram to show the player how the network is set up.

The game also has a character called Colonel Whiskers to introduce new concepts to the player. This character starts by looking down on the player, but gradually begins to acknowledge the player's achievements and becomes more friendly.

Upon analyzing the memory and CPU usage of the game, an estimate of the requirements were derived. The game needs 1.5 Gb of ram, as it can reach 1.03 Gb of RAM

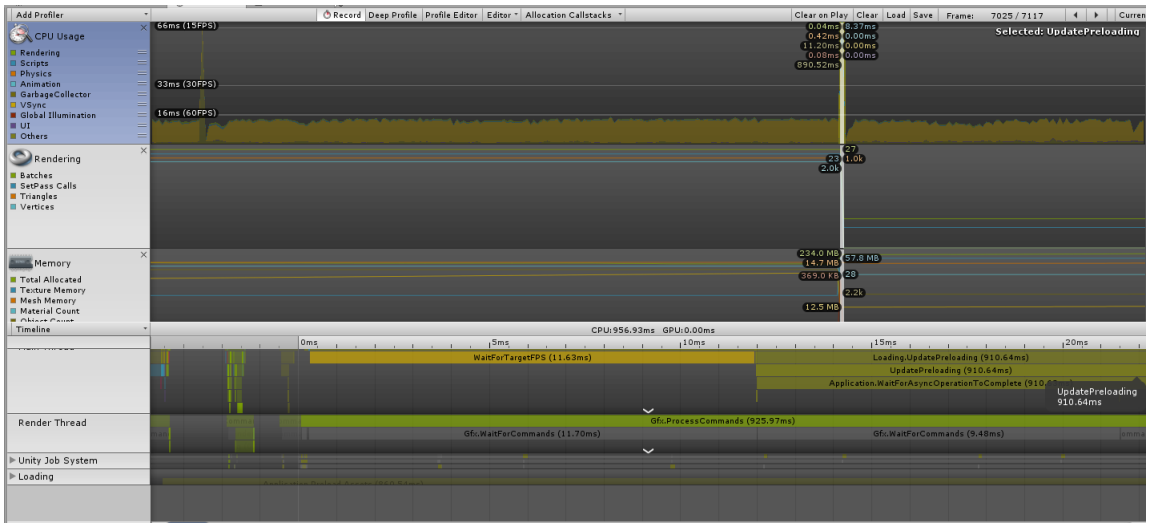
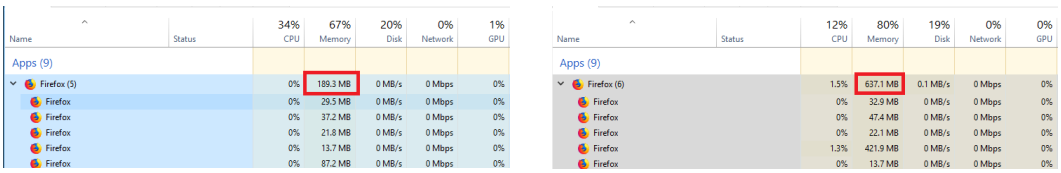


Figure 5.8: Unity Profiler showing CPU usage



(a) RAM usage of Firefox before running the game

(b) RAM usage of Firefox after running the game

Figure 5.9: RAM usage of WebGL build of the game

usage maximum. We could not conclude a minimum requirement for the CPU but we analysed the usage of the game within the Unity game engine to analyze how long the longest task of the game can take.

Chapter 6

Evaluation Methodology

This chapter describes the different techniques used for the evaluation of the game. We discuss the pre-game and post-game questionnaire; what they are and how they help us assess the teaching capabilities of the game. We also run a session called a think-aloud to understand users' interactions and thoughts while playing the game. Lastly, we talk about the System Usability Scale; what it is and why it is used to evaluate the usability of this game.

6.1 Pre-game questionnaire

The pre-game questionnaire is attached in Appendix D. The questionnaire was given to the participant before they began to play the game to assess their prior knowledge. The questions were designed to be similar to the post-game questionnaire.

6.1.1 Section 1

This section tested if a participant fell into the target group for the game, i.e. they have not taken computer security and worked with the command line before playing the game. I ask them to say how often they play video games in a week in an attempt to find a correlation if participants who played video games sometimes would do better at my game, i.e. progress through the levels more quickly and with less difficulty.

6.1.2 Section 2

The participants are asked to rate their current knowledge on `iptables` to see how they feel about it themselves. Question 2 is to assess if the participant knows about protocols and into how much detail.

The third question is to assess their knowledge on a simple `iptables` command. This

question consists of several parts, each asking about the different parameters the command can take. This split is done to see if the participant can understand certain properties of the rule simply by reading it. In this way, we can assess what areas the participant is good in as well as what areas the participant would need to learn to understand the full command of `iptables` e.g. the participant might know that 'ssh' and 'tcp' refer the protocols.

6.2 Think-Aloud

A think-aloud is a research method in which participants speak aloud any words in their mind as they complete a task [34]. The participant was read a script for this session, which is attached in Appendix F. This script does some initial training of the think-aloud for the participant to know how they should behave during the study. It was ensured that the think-aloud was held in a quiet area where they would not be distracted. These were also audio recorded for myself to listen to later and check if there was anything I forgot to write during the think-aloud. The script was taken from the Human-Computer Interaction course at the University of Edinburgh and edited to my requirements with permission.

The purpose of this think-aloud was to see how users navigate around the interface of the game. It would also allow me to understand if any my game lacked an explanation of some concepts, which I could use to improve those explanations.

6.3 Post-game questionnaire

This questionnaire is in Appendix E. This questionnaire was given to the participant after they finished the game to assess if they had learned what the game aimed to teach them.

6.3.1 Section 1

These questions ask the participant what they generally felt about the game. Also, I asked the participant what level they reached in the game in case the participant wished to leave early. There is a question asking the participants to say if the game was fun and easy to navigate as well as how engaging they felt the game was. Suggestions and improvements are also asked to get feedback on what the participants liked about the game and what could be improved upon.

6.3.2 Section 2

These questions are to evaluate the participant on what they have learnt from the game. The first question asks them what they think the command does which is to evaluate if they have learned the syntax and what this command would do in a firewall. This question is no longer in parts as I wanted the participants to know the full command by now and what each part meant. The next questions are based on firewalls which are shown using the game's interface used in the level of the game. These last two questions are there to assess if the player understands firewalls rules properly, which means if they understand the **INPUT** and **OUTPUT** chains as well as the ordering of rules. It is also to assess if the player remembers that application protocols use internet protocols such as TCP. The last question also intentionally asks "If Bob was hosting a web server on the default port" to assess if the player remembers that HTTP is used for web servers in this case. In addition, I assess if they remember that SSH requires that the firewall allows traffic in both **INPUT** and **OUTPUT** chains.

6.4 Experiment Details

The participants played the game after it was compiled to a WebGL format in a Google Chrome browser. They were also given the pre-game questionnaire at the start of the game and the post-game questionnaire at the end of the game. When the session began, I informed the user what a think-aloud is, as shown in the script attached in Appendix F. In this script, I call it a talk-aloud so that participants are not confused and talk more often.

6.5 System Usability Scale

The System Usability Scale, or SUS, was invented by John Brooke in 1986 and is a quick form of an evaluation tool to evaluate a wide range of services which includes software [35]. The SUS is a valid system to differentiate between a usable and unusable system, which is why I decided to use this.

On the project demo day at the University of Edinburgh that was held in the Informatics Forum, I constrained the game only to have two levels. After these two levels, the participant was shown the winning screen and asked to fill in the System Usability Scale questionnaire, attached in Appendix G. The first two levels go over most of the game's usability features and thus is enough to evaluate only the usability of the game.

6.6 Summary

Different techniques were used to evaluate the usability and teaching capabilities of the game. A pre-game and a post-game questionnaire were created to assess a participant's prior knowledge against their knowledge on firewalls after playing the game. Besides this, a think-aloud session was conducted to analyse a participant's thoughts and interactions while playing the game. Lastly, to assess the usability of a proper scale, the System Usability questionnaire was used on the project demo day at the University of Edinburgh. For this session, the participants were only shown two levels of the game, as this was purely done to assess the game's usability. Given that the levels look similar to one another, this was sufficient for this evaluation.

Chapter 7

Results and Evaluation

This chapter analyses the results achieved by evaluating different participants. We initially discuss the iterative design aspect of the game, how it was before evaluation, what was changed after evaluation and why. The game was first evaluated on two experts and altered according to their feedback. We will then assess the game on our target group to test the teaching capabilities of the game. Finally, we present the results of the System Usability Scale.

7.1 Iteration

The game has gone through iterative development, i.e. some features and elements of the game have been altered according to the feedback from evaluation. There have been two total iterations of the game.

7.2 First Iteration

7.2.1 Game Design

The game previously had different UI elements in the level screen, as shown in Figure 7.1. The buttons for the briefing panel and to reset the chains were different. This level also did not have numbers for the rules; hence, when the player had to delete a rule, they would have to manually count the rules to know the line number that is required for the command to delete a rule. The button to submit the rules for that level was called **Play** instead of **Submit** in this iteration.

Another feature this iteration did not have was the capability to run a command from the terminal by pressing **Enter** on the keyboard instead of pressing the **Run** button on the screen. In addition to this, the game did not have proper error checking to inform the user what specific keywords in their command were wrong. Instead, it told the user

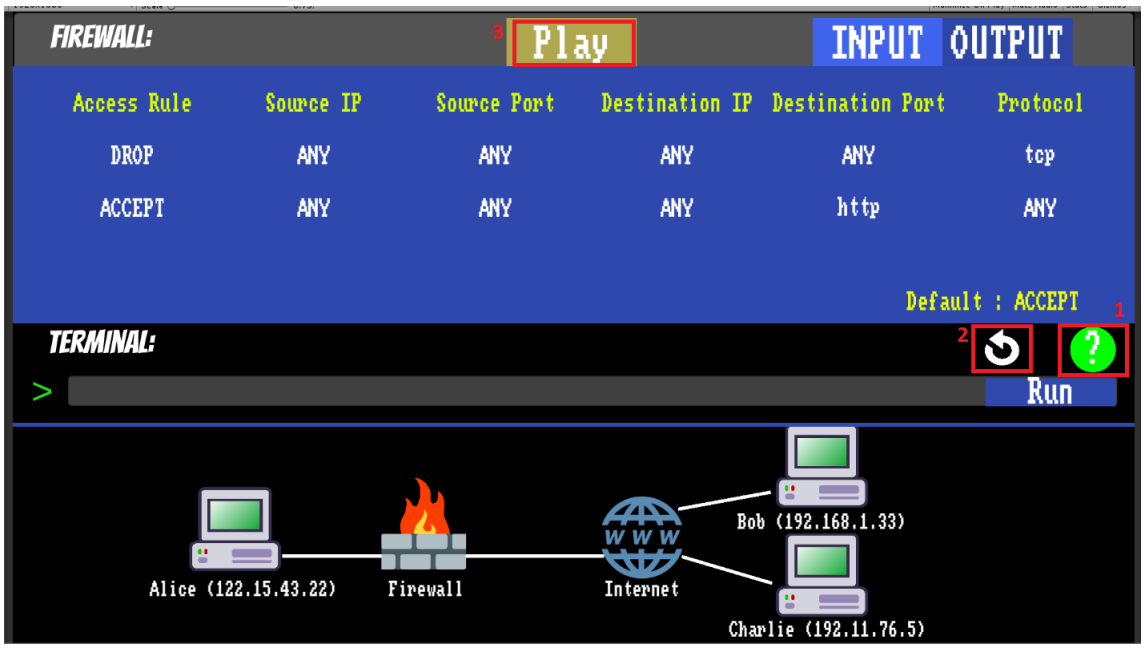


Figure 7.1: The first iteration of the game, red boxes showing the different UI elements, where 1 is the button to show the level briefing, 2 is the button to reset the chains, and 3 is the button to submit the rules for the level

that the command was wrong and to try again.

Also, this iteration of the game did not teach its players how to delete a rule until the very last level of the game, which in the final iteration is the third level of the game.

7.2.2 Evaluation with Experts

Expert evaluations involve using the experience of experts in the field to analyse potential problems in a design solution. Expert evaluations are a powerful tool to assess the user experience of a system and identify technical issues before more costly user studies take place. As compared to normal user studies, they can evaluate material regarding technical issues and give suggestions based on how to improve the usability of a system.

7.2.2.1 Method

Two experts evaluated the first iteration of the game. These were undergraduate students in their fourth year of Computer Science at the University of Edinburgh.

Given that they had taken the Computer Security course and also taken courses on usability such as Human-Computer Interaction, their feedback was vital to improve the game's usability. In addition to this, they were both doing research projects that

focused on usability and learning. These experts spent a considerable amount of time evaluating my game, giving suggestions and explaining different ways the game could improve.

Expert 1 was also doing a usability-focused project, working alongside PhD students in the University's Usability Lab. Thus his feedback was vital in assessing the usability of the game. Expert 2 had a project focusing on educational games and using Unity. Therefore his feedback was essential for evaluating the learning and gaming aspects of the system.

7.2.2.2 Expert 1

The expert gave detailed feedback on different elements while playing the game. He commented that while the text of the briefing by the Colonel was a lot to read, the addition of personality into the text from Colonel Whisker helped engage him in reading it. He found the UI tutorial quite informative from a usability point of view, as it covered all the vital elements that were required by the user. The expert liked that Colonel Whiskers personality changed as the game progressed and said this was a good aspect to keep the players engaged.

However, this expert did notice some issues with the game. In the briefing panel, the text slowly loads letter by letter to simulate an easier to read format and as to not show too much information immediately. He suggested that this slow loading of the text could make players impatient, and an option to load all of the text quickly when he presses the **Next** Button while it is loading would be a good feature to avoid this issue. He also suggested changing the text of the **Play** button to something else, as some players might be confused given that they were already 'playing' the game.

Also, he said that using a button with a question mark on it as the briefing button in the level might confuse players into thinking that this is the hint button, and hence might not press it if they felt challenged. He also noted that the level to teach the player how to delete a rule should not be at the end of the game, as a player might prefer to delete rules while playing the game. Another suggestion was that the player should be allowed to add a rule from the terminal by pressing the **Enter** key on the keyboard instead of clicking the **Run** button all the time. The button to reset the chains could also be confusing, as he said that players might think it meant to undo their last action. Also, the button to reset the chains was too close to the briefing button, which could lead to accidentally clicking the wrong button.

The suggestions from this expert were vital to improving the user interface of the game. It pointed out direct ways to make the system more user-friendly, which are taken into consideration to enhance the game for the second iteration.

7.2.2.3 Expert 2

The expert was invited to play the game. He said that the game presented well-balanced challenges, ones that were not too hard but not too easy. He liked the idea of having a terminal input command to simulate the actual configuration of firewalls.

However, he commented on several issues with the game. He was not fond of the explanation of some concepts in places, e.g. he said players would need a better description of what a policy meant and more detail on ports. While he knew about `iptables` commands, he said that the game should inform the player that they were case sensitive, as they might assume that this is not the case. The **Play** Button also confused this expert for the same reason as the first expert, i.e. the player would already be 'playing' the game. He also suggested I present the tasks for the level in a more readable and easy-to-follow manner, e.g. in numerical order so the player could keep track of what the player had done so far and what he has yet to do. Also, he suggested that while he already knows what chains were, adding in a visual aid would assist in identifying which direction of packet movement the **INPUT** and **OUTPUT** chains handled.

7.2.3 Conclusion of Expert Evaluation

The experts provided valuable feedback on how to improve the game from a usability, educational and gameplay point of view. From this, I changed the following aspects of the game:

- Teach a player how to delete a rule earlier (Now at Level 3)
- Better explanations about what a policy is and what a port is with a little more detail.
- Change the position of the button to reset the chains, and instead of an icon, made it into a button with text that said 'Reset Chains'.
- Added a separate **Hint** button and a **Briefing** button to assist players.
- Improved briefing of the level to show what the player has to do in a numerical list format.
- Pressing the **Next** Button in the briefing with Colonel Whiskers while the text is loading would load all of the text instead of going to the next instruction. The player could then go to the next instruction by pressing the **Next** Button again.
- Hitting the **Enter** button on the keyboard will run the command if the terminal input field is focused.
- When the Colonel talks about the **INPUT** and **OUTPUT** chains to the player, his image is with a diagram, and an animation to show packet movement into or out of the network, to show which direction of packet movement each chain manages.

After these changes, I proceeded to test this game on my target audience further.

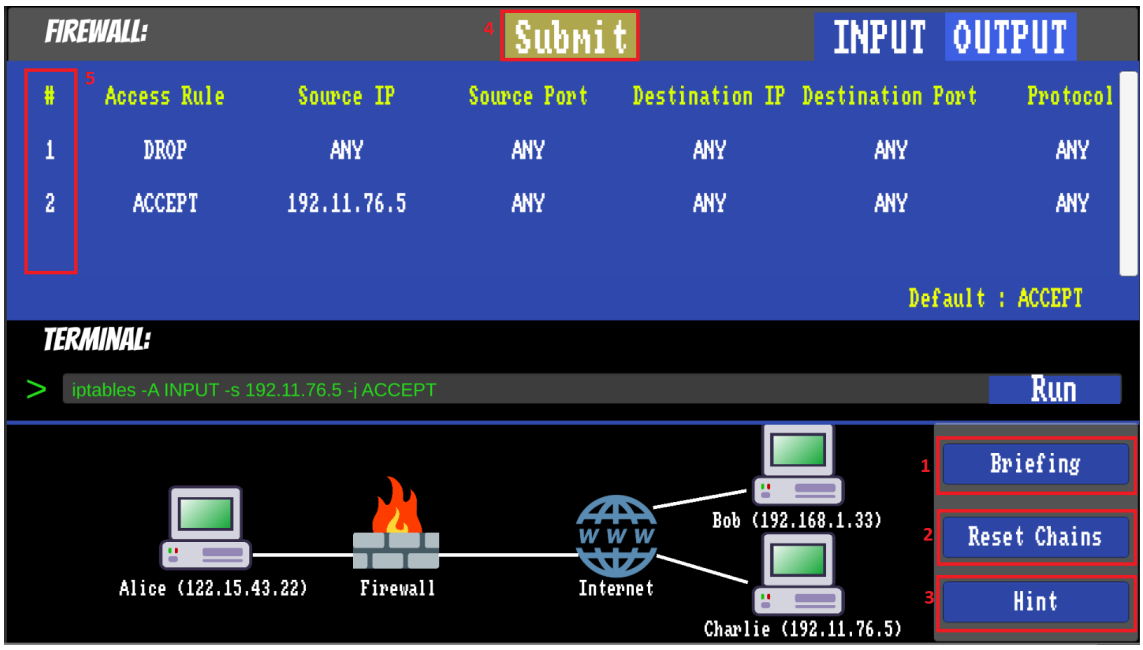


Figure 7.2: A level after rules are added. Red box 1 shows the new button to show the briefing. Red box 2 showing the new button to reset the chains. Red box 3 shows the new hint button to assist the player in case they are stuck. Red box 4 at the middle top shows the button that now says 'Submit' instead of 'Play' to not confuse players. Red box 5 shows the line numbers for each rule to assist the player in deletion

7.3 Second Iteration

After patching in the changes mentioned in Section 7.2.3, the level now looked like the final design as shown in Figure 7.2. Six participants in total evaluated this game, which excludes the experts. Figure 7.3 summarises the details of these participants. Participant 4 (P4) had taken Computer Security but said he wanted to learn about `iptables` again as he had forgotten most of it, so a study was organised with this participant as well.

7.3.1 Results

7.3.1.1 Pre-game questionnaire

The participants were grouped based on what specific mistakes they had in their pre-game questionnaire. This grouping was done to make it easier to analyse and evaluate their learning after they filled in the post-game questionnaire.

Figure 7.4 shows common answers from the participants. The participants who partially got the syntax correct got some parts wrong, e.g. none of the participants knew what '-j DROP' stood for, but they guessed that '-s 192.154.12.34' meant the source IP address and '-p tcp' meant the type of protocol. Participant 4 (P4) got all of the syntax

Participant	Year	Taken Computer Security?	Plays video games often?	Knowledge on 'iptables'?
P1	UG4	No	Rarely	Never heard of it
P2	UG3	No	Sometimes	Heard of it but never used it
P3	UG4	No	Never	Used it a few times
P4	UG4	Yes	Sometimes	Used it a few times
P5	UG4	No	Always	Heard of it but never used it
P6	UG2	No	Sometimes	Never heard of it before

Figure 7.3: Participants List

Participants	Answers
P2, P6	Didn't know 'iptables' syntax at all
P1, P3, P5	Partially got the syntax correct
P4	Got all of the syntax correct
P2, P6	Ticked only one more box than needed
P3, P5	Ticked two or more boxes than needed
P1, P4	Ticked only the correct boxes

Figure 7.4: Pre-game Questionnaire Results

correct as he had taken the Computer Security course before this and did recall some aspects of the syntax.

The second question was categorised by which boxes the participant ticked. Ticking one more box than needed is not a major problem as the options could be confusing to people. However, ticking two or more wrong boxes meant that the participant might not know about HTTP as the other options are not commonly done using a web browser. Using the results, I further categorised the participants based on their prior knowledge on 'iptables' and protocols, as shown in Figure 7.5.

The participants were further grouped to assess their overall knowledge in compari-

Knowledge level	Participants
Very little knowledge of 'iptables' and protocols	P2, P3, P5, P6
Some knowledge of 'iptables' and protocols	P1
Good knowledge of 'iptables' and protocols	P4

Figure 7.5: Participants final Grouping based on prior knowledge

son to other participants. Participant 1 (P1) got the syntax partially correct but also ticked only the correct options for HTTP, which suggests he has better knowledge of HTTP as compared to other participants. Participant 4 (P4) had good knowledge as he had taken the Computer Security course before this, making him the participant with the best knowledge before playing the game.

7.3.1.2 Think-aloud

Open coding has been used to analyse the data from the think aloud. Open Coding includes labelling concepts, defining and developing categories based on the properties and dimensions of the data [36]. It is used to analyse qualitative data and part of many Qualitative Data Analysis techniques. The data can be grouped using open coding and abbreviated to reach conclusions based on that data.

From the open codes of data, I was able to categorise the data into themes. Figure 7.1 shows these themes. Overall, there were many features that the participants liked about the game. All participants also read the text of the game carefully which ensured a smooth think-aloud with fewer issues. The participants said that the comedic aspect of the Colonel made the instructions for the game more fun to read. Some features and improvements were suggested, such as adding in lives to the game to limit the number of times a player can get a level wrong. This feature could make some participants give up on the game if they could not get through a level after some time, so I did not consider adding such a feature. However, this would be possible by allowing a player to select the level of difficulty before playing the game to challenge players who want to be so.

Participants did enjoy the different ways to facilitate their learning in the game, such as the animations for the **INPUT** and **OUTPUT** chains. During the game, the participants also remembered what each protocol was. There were, however, some syntax errors made by participants. These were small errors that were quickly fixed by all participants. Some participants started thinking of new parameters that were not taught to them, such as a destination IP address for the command. They did this when they could not get through a level. While setting the destination IP address is a valid parameter for `iptables`, it is not covered within the scope of the game. While the participants only tested out these new parameters and removed them after testing them once, the player must be given clear indications that they do not require any additional knowledge other than what the game has taught them.

7.3.1.3 Post-game questionnaire

The post-game questionnaire is designed to assess the participant based on what the game has taught them. The results were overall quite positive with most participants answering most of the questions correctly. The questionnaire is attached in the Appendix E.

Open Codes	**Themes**	
Enjoyed game and liked Colonel Whiskers; Fond of Colonel Whiskers change in personality; Liked the syntax remembrance; Liked the prize; Clicked other buttons in the main menu other than just the 'Play' button	Features participants enjoyed	Game positives
Did not find the game too difficult; Chains were well explained	Aspects of the game participants found simple	
Tendency to press hint right away; Read text carefully	Notable interactions of participants in the game	
Better understanding of chains after INPUT and OUTPUT animation; Forgot the opposite of DROP while playing but remembered when seeing that the default policy said ACCEPT; Participants started remembering what protocol does what	Good elements in the game to facilitate learning	
Ability to copy and paste a command; Player has a limited amount of times he can get an answer wrong until it's game over	Suggested Improvements	Game Improvements
Syntax issues e.g. case sensitivity issues; Trying random parameters in the command that were not taught by the game;	Some explanations within the game needed to be improved	Player issues

Table 7.1: Think aloud themes derived from open codes

Participant	Found game fun?	How easy was game to play?	Rate Engagement from 1 to 10	Was game easy to navigate?
P1	Yes	Somewhat easy	10	Yes
P2	Yes	Fairly easy	10	Yes
P3	Yes	Somewhat easy	7	Yes
P4	Yes	Fairly easy	7	Yes
P5	Yes	Somewhat Easy	8	Yes
P6	Yes	Not Particularly easy	8	Yes

Figure 7.6: Post-game section 1 results for each participant

Section 1
This section aimed to get the participants general feedback about the game as well as any suggestions they might have to improve the game. All participants reached the final level of the game. Figure 7.6 shows the answers of the participants to the questions.

All of the participants found the game both fun and easy to navigate. In terms of the game's difficulty, one participant said that it was not particularly easy, while every other participant found the game either fairly or somewhat easy. This game was generally designed to be somewhat challenging, so having the game be somewhat easy was the intention. P6 specifically did not find the game particularly easy. This participant was the only participant from the second year of Informatics, whereas all other participants were senior, which might be a possible reason. In terms of engagement, all players found the game quite engaging, with P3 and P4 giving the lowest score of engagement as 7.

Most participants favourite part of the game was Colonel Whiskers. They found him both amusing and interesting. Another part of the game that appealed to the participants were the animations for the **INPUT** and **OUTPUT** chains to make clear which chain handles which packet when concerning the network. It was suggested to increase the learning through animations rather than text for the game. P5 and P6 found the user interface of the game quite interesting and well themed. All participants also enjoyed the game music and felt it enhanced their in-game experience. P5 said that the prize of the game was quite nice, although he is the only one who liked it. Participants commonly suggested allowing the feature to copy and paste the command from the briefing panel rather than write it all down manually. This feature was one that I did not wish to implement, as I believed it might hinder learning in the aspect that they would simply be copying off the command and editing it rather than type it out and think of each parameter to change.

Section 2
This section was to evaluate if the participants had learnt everything listed in the learning goals. Figure 7.7 shows the result of the questions for each participant.

The results are overall positive, with every participant answering the first question correctly, i.e. they learnt the syntax of 'iptables' properly and could describe everything

Participant	Question 1 (iptables syntax)	Question 2 (a) (INPUT chain and SSH)	Question 2 (b) (INPUT chain and SSH)	Question 3 (a) (OUTPUT chain and HTTP)	Question 3 (b) (OUTPUT chain and TCP)
P1	Correct	Correct	Correct	Wrong	Correct
P2	Correct	Correct	Correct	Correct	Correct
P3	Correct	Correct	Correct	Wrong	Wrong
P4	Correct	Correct	Correct	Correct	Correct
P5	Correct	Wrong	Correct	Correct	Correct
P6	Correct	Correct	Correct	Correct	Wrong

Figure 7.7: Post-game section 2 answers for each participant

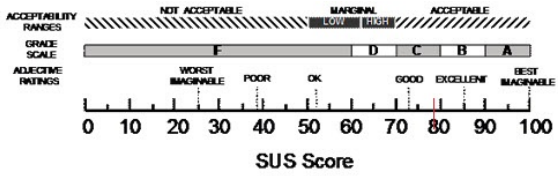


Figure 7.8: A comparison of the adjective ratings, acceptability scores, and school grading scales, in relation to the average SUS score, taken from [38]. The red line indicates the score achieved by the game in this project

in the command correctly. Given this, the game successfully taught the participants the syntax of 'iptables'. P5 got question 2 (a) wrong. When telling them of the actual answers, they said that they forgot that it was the **INPUT** chain being assessed in the question as they forgot to read it. Question 2 (b) was also answered correctly by all participants.

Question 3 (a) seems to be the trickiest question, as half of the participants got it wrong. They said that they forgot that HTTP uses TCP, and answered this by selecting 'Cannot Say' or that Alice would be able to access it. Due to the number of participants that struggled with this, it suggests that the game needs to improve its explanation on this particular concept. While most participants got Question 3(b) correct, two got it wrong. This was due to a similar reason with HTTP, where they forgot that SSH uses TCP.

7.3.1.4 System Usability Scale

There were 16 participants who played the game during the demo day and responded to the survey. After calculating the score through the method suggested by Nathan Thomas in his usability guide [37]. Upon calculation, my system obtained a score of 79.6, which is above the average score of 68 for the SUS. After adding an adjective rating to this score, it has been placed in the good category for usability, as shown in Figure 7.8.

7.4 Evaluation Conclusion

It was defined Section 4.3.3 the learning outcomes expected out of this game. Going through them again:

- **The syntax of `iptables`** : All participants got the question related to the syntax correct in the post game questionnaire. On this basis, I would say that given the parameters specified in that question, all participants could simply answer it completely.

- **The INPUT and OUTPUT chains** : This is related to both Question 2 and Question 3. Given that the reasons by some participants who answered some parts of the question wrong was that they either misread the question or forgot about a learning goal other than chains, I believe within these participants, the game taught them about the INPUT and OUTPUT chains well.

- **TCP and UDP, the basics enough to understand firewall rules** : The game needs to improve on this aspect. Participants who got Question 3(b) wrong said that they forgot that HTTP used TCP, which is what the game teaches them. Due to them not remembering this after playing the game, I believe the game needs to emphasize this more and have a more practical level based on this rather than the one already present. UDP is not assessed in the post-game questionnaire, hence no conclusion can be reached about how well UDP is taught in the game.

- **The application protocols HTTP and SSH, just the basics to understand firewall rules** : This learning goal was also assessed by question 3. While a few participants got this question wrong, they did say that they could clearly identify that a web server on a default port would be the port of HTTP. They also said they knew that SSH required its packets to be allowed both ways. This might suggest that HTTP and SSH were taught well, it was due to the fact that they misunderstood TCP that they failed to answer the question properly.

- **The different ports regarding the different application protocols** : This goal is less about having the participants remembering each port for each protocol but rather more about them knowing that each application protocol has its own default port number associated with it. This was a difficult goal to assess as there is no direct question I could come up with for this. However, from the fact that they understood what 'web server on default port' meant in Question 3 (a), it might suggest that they had no issue associating these protocols to ports.

The game also received a good SUS score, which shows that it is usable. However, this score can still be improved upon which might involve more research into user interface design.

7.5 Summary

The game successfully taught the players most of the learning goals. However, TCP and UDP need a better explanation for the players to understand about them. Explaining the OSI model might assist some players to see the relationship between the protocols. Only two participants were quite fond of the prize in playing the game. Given that this was supposed to be a reward for winning the game, more innovative rewards would lead to higher player satisfaction on completion of the game.

Chapter 8

Conclusion

8.1 Overview

The aim of this project was initially to create a game to teach informatics students about firewalls and `iptables`. We created an initial game for this purpose after researching different ways to engage users. However, after two participants played this version of the game, they had multiple issues with navigation. Despite a user interface tutorial, excessive elements present within the interface still lead to confusion. In addition to this, we realised that the target audience was too broad, as different informatics students would have different experience levels with computing concepts, and hence require different ways to explain concepts.

Given this, we decided to focus on creating an educational game dedicated to teaching the concepts of firewalls, protocols and `iptables` introduced in the Computer Security course at the University of Edinburgh. To gather requirements, we created a survey hosted on Google Forms and sent this out to students who had previously done Computer Security. In addition to this, we asked people questions directly if they wished not to do the survey. We asked them about the difficulties they faced in the course, why they lost marks on the questions related to `iptables` and what tools they would have found helpful to assist them in learning about `iptables`. From their responses, we extracted requirements, and learning goals were formed based on these requirements. We decided to have the game teach a subset of the content taught by the course related to `iptables` and conduct user studies based on a game teaching only this subset of concepts, expanding the game in the future after evaluation.

We then created the game based on these learning goals, which consists of eight levels. Each level teaches a different concept to the player and challenges them based on those concepts. To properly evaluate this tool, we first conducted an expert evaluation with two experts to ensure common user issues could be avoided before conducting user studies with the target audience. The feedback provided by the experts led to a second iteration of the game with several improvements. User studies were then conducted with the target group using a pre-game questionnaire and post-game ques-

tionnaire. The usability of the game was also assessed by having participants play the game while participating in a think-aloud session. To further evaluate the usability of the game, we used the System Usability Scale.

8.2 Results achieved

Given the evaluation, we concluded that the game was indeed a useful learning tool for teaching the learning goals of this project. Most of the participants successfully learnt the syntax of `iptables` and about the application protocols. They also understood the concept of chains well.

However, some participants did not fully understand some concepts which could be improved upon as seen in Chapter 7. For more in depth evaluation for the game, a large-scale user study should be carried out.

8.3 Future Work

As the game only teaches a subset of the material taught by the Computer Security course, scaling the game to teach the remaining concepts is something that can be done in the future.

The game currently has a lot of text to explain different concepts. While some text within the game is imminent, the addition of animations and aesthetics could divert players' attention to different things, thus possibly aiding them in reading without realising they are learning something. More research is required to discover different ways games can succeed in conveying information to users while keeping them engaged.

The addition of the Colonel Whiskers character engaged users with its comedic personality. Having more characters with different personalities might further engage players in playing the game even more.

The winning prize of the game was received well, where many users enjoyed the video. However, a more satisfying reward for finishing the game might enhance the experience for players, e.g. a high score.

Appendix A

Game Images

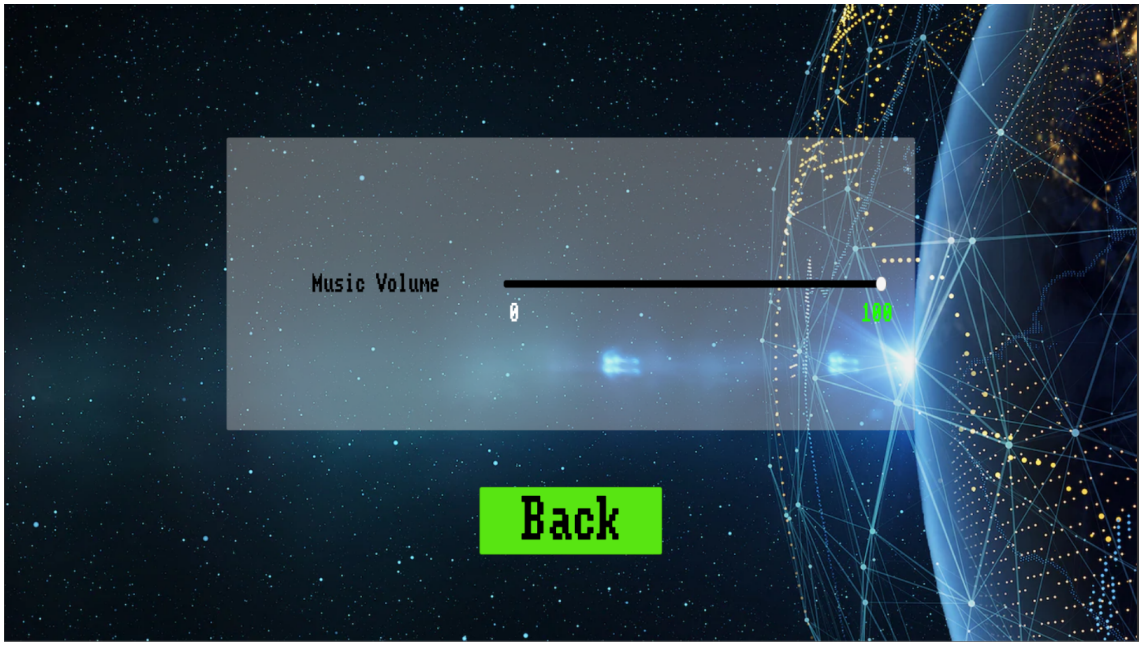


Figure A.1: Options Screen of Game

Figure A.2: About Screen of Game

Figure A.3: Credits Screen of Game

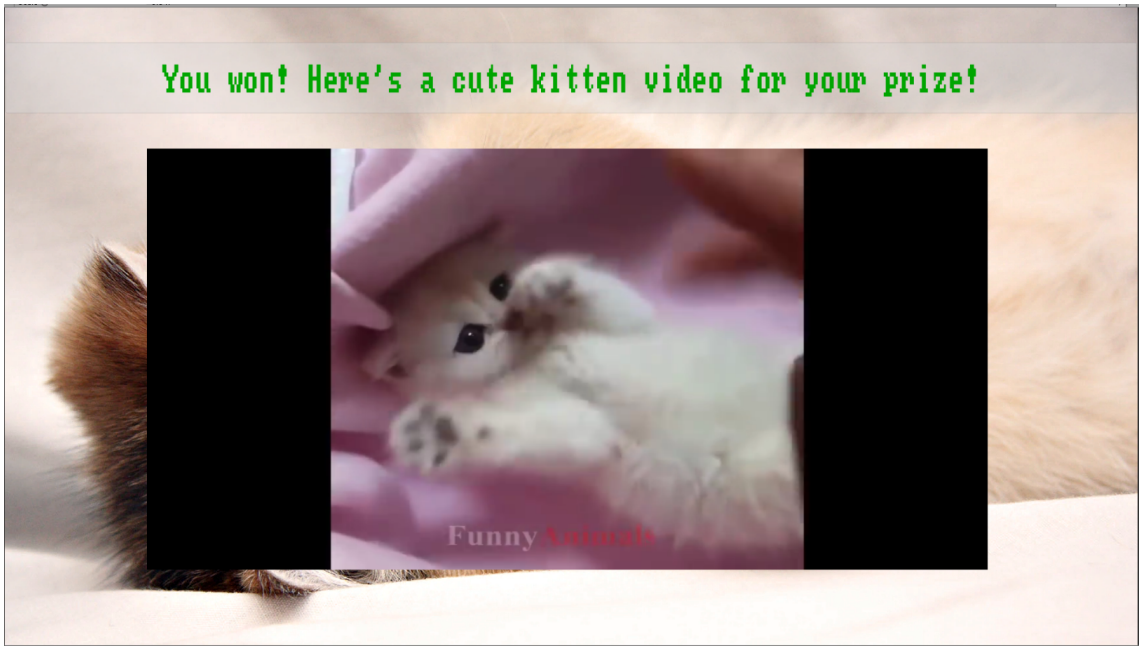


Figure A.4: Winning Screen of the game

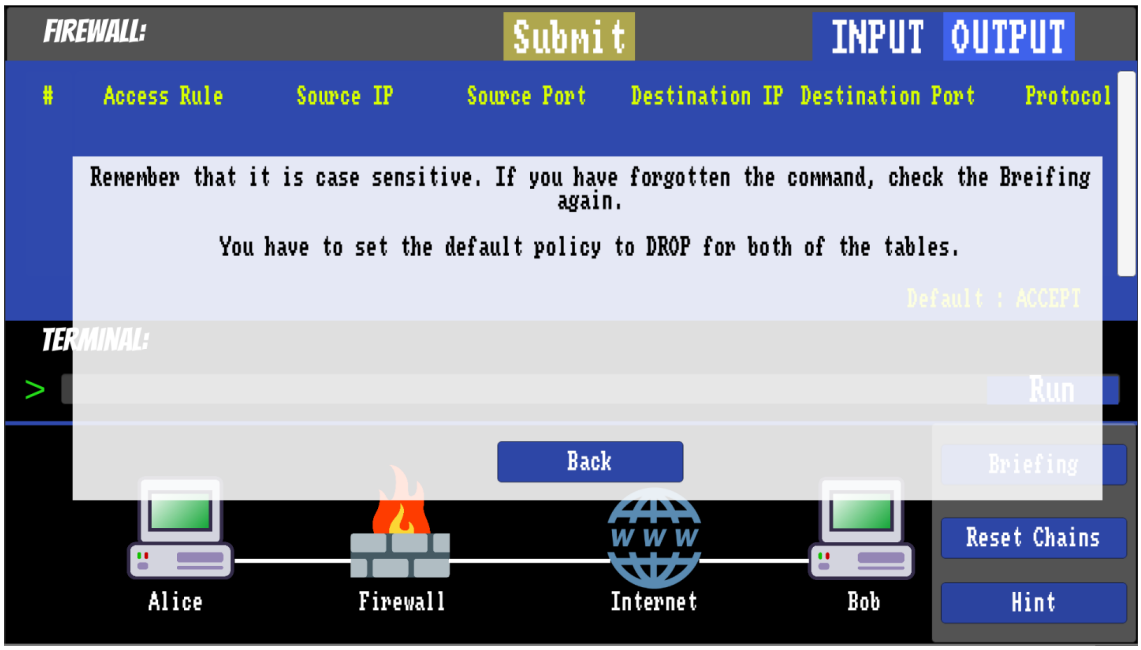


Figure A.5: Hint Screen of first level

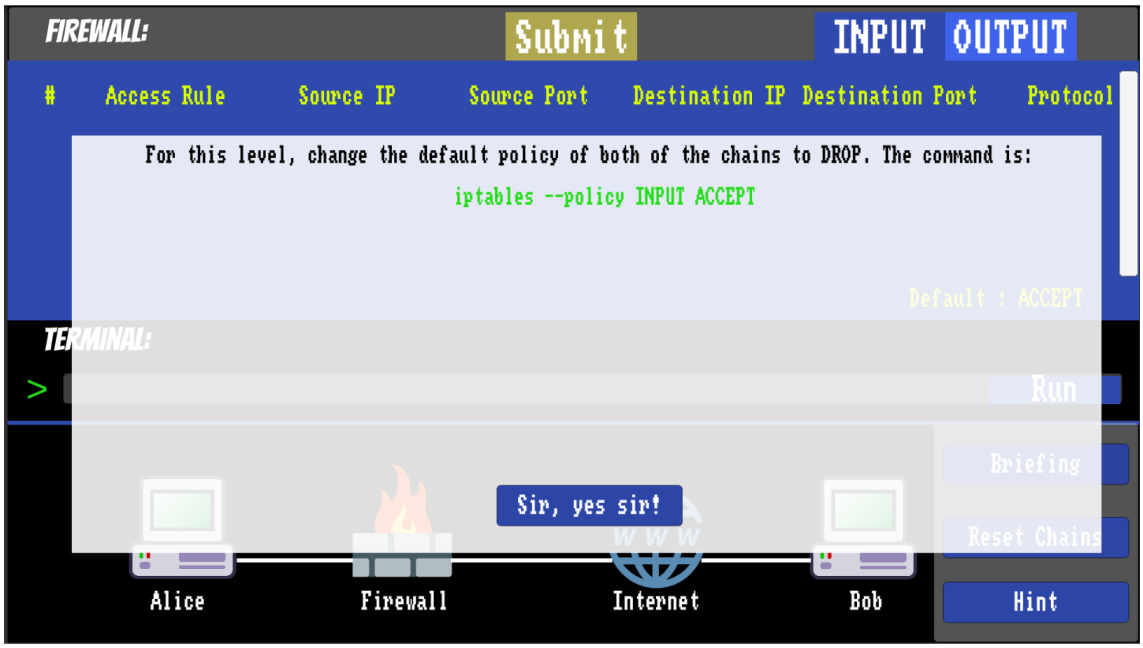


Figure A.6: Briefing Screen of first level

Appendix B

Requirements Gathering Google Forms Questionnaire

Computer Security 'iptables' Survey

This survey is intended to gather information of what people found hard about the 'iptables' section of the firewalls in the Computer Security course taught at the University of Edinburgh.

This study will be used to gather requirements for the Firewall game, any feedback you provide will be used for this purpose along with anonymous use in related work. The data collected from this survey will be kept for a maximum of two months and then destroyed. Anonymized quotes may be retained longer for use by future students on this project.

* Required

1. **Email address** *

2. **What is your name?** *

3. **Are you a student at the University of Edinburgh studying in the School of Informatics?** *
   Mark only one oval.
   - Yes
   - No *After the last question in this section, stop filling out this form.*

4. **How would you describe your Computer Security experience?** *
   Mark only one oval.
   - None
   - Little
   - Average
   - Above Average
   - Expert

5. **Did you take the Computer Security course in the past?** *
   Mark only one oval.
   - Yes
   - No *Stop filling out this form.*

Coursework Experience

This study will be used to evaluate the difficulties faced by students in the coursework for the design of the Firewall game, any feedback you provide will be used for this purpose along with anonymous use in related work. The answers will be kept for a maximum of two months and then destroyed. Anonymized quotes may be retained longer for use by future students on this project.

6. **What grade did you get in CS CW1?** *

7. **How many marks did you get on the firewall question in CW1 (QS 27-29)?** *
   (You can view your mark scheme on the Learn website of CS, and then click on "My Grades" on the left, then click on "Coursework 1", and then click on your mark under the column "Calculated Grade" ")

8. **What did you get in question 27?** *

9. **If you lost marks in question 27, what was your feedback? (Just type in N/A if you did not)** *

10. **What did you get in question 28?** *

11. **If you lost marks in question 28, what was your feedback? (Just type in N/A if you did not)** *

12. **What did you get in question 29?** *

13. **If you lost marks in question 29, what was your feedback? (Just type in N/A if you did not)** *

14. **What, if anything, did you find hard about those questions?** *

Experience with Iptables

15. **Did you feel you had a sufficient knowledge on the subject before beginning the coursework?** *
    Mark only one oval.

    - Yes
    - No

16. **What, if any, further resources did you use to learn about 'iptables'?** *

17. **How strong, if at all, do you find your knowledge on 'iptables' as of now?** *
Rate it from 1 to 10.
Mark only one oval.

1	2	3	4	5	6	7	8	9	10
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

18. **Do you think an additional information on Firewalls such as a game would have helped you in the coursework?** *
Mark only one oval.

- ◯ Yes
- ◯ No

Powered by
Google Forms

Appendix C

Consent Form

‘Firewall Learning Game’ Consent Form

This project aims to gather information about the usability of a game intended to help users learn about Firewalls.

Today we will be doing a talk aloud study alongside a brief survey about your thoughts on the usability of the given Firewall game. During this talk aloud, I will show you the application and ask you to evaluate it given some outlined scenarios. I would then appreciate if you could provide your general thoughts and opinions about the application based on your experiences with it, on the accompanying survey.

We will be audio recording during the study. If you feel uncomfortable about this at any time, you may stop the recording or tell me that the next bit should not be quoted. There will not be any compensation for participation in this study beyond the knowledge that you will have advanced science.

This study will be used to evaluate and improve upon the designed User Interface for the Firewall game, any feedback you provide will be used for this purpose along with anonymous use in related work. The audio and transcripts will be kept for a maximum of two months and then destroyed. Anonymized quotes may be retained longer for use by future students on this project.

The project is supervised by Dr Kami Vaniea (kvaniea@inf.ed.ac.uk) and conducted by myself, Rusab Asher (s1572691@sms.ed.ac.uk).

[ ] I understand that I am participating in a study as part of the “Vulcan - The video game to teach Informatics students about Firewalls” project.

[ ] I am willing for the audio to be digitally recorded and transcribed as part of this research project

[ ] The researcher may use **audio/ literal quotes** from the study provided that the quotes are anonymised and cannot be connected back to me.

Participant: _____ Date: _____
Researcher: _Rusab Asher__ _____ Date: _____

Appendix D

Pre-game Questionnaire

Pregame questionnaire

Section 1

1) What school of the university do you belong to (if any)?

   _____

2) What year of university (if any) are you currently in?

   _____

3) How often do you play video games in a week?
   - ☐ Never
   - ☐ Rarely (1-2 times a week)
   - ☐ Sometimes (3-4 times a week)
   - ☐ Often (Daily once or twice)
   - ☐ Always (Multiple times a day)
4) Have you ever taken a course of Computer Security before?
   - ☐ Yes
   - ☐ No
5) Have you worked with the command line before?
   - ☐ Yes
   - ☐ No

Section 2

1) How would you rate your knowledge on 'iptables'?
   - ☐ Never heard of it before
   - ☐ Heard of it but never used it
   - ☐ Used it a few times
   - ☐ Use occasionally every now and then
   - ☐ Use it often

2) Please tick all the boxes in which you think a computer would commonly use HTTP to accomplish these:

   a) Remotely logging into a computer ☐

   b) Streaming an online video ☐

   c) Sending a file online ☐

   d) To pet a kitten ☐

   e) To open your file explorer ☐

   f) Opening a Facebook event page ☐

3) Consider the following command to form a rule in a firewall using 'iptables'

iptables -A INPUT -s 192.154.12.34 -p tcp –dport ssh -j DROP

a. What do you think '**-A INPUT**' means?

_____

_____

b. What do you think '**-s 192.154.12.34**' means?

_____

_____

c. What do you think '**-p tcp** means?

_____

_____

d. What do you think '**---dport ssh** means?

_____

_____

e. What do you think '**-j DROP'** means?

_____

_____

Appendix E

Post-game Questionnaire

Postgame questionnaire

Section 1

1) What level did you reach at the end of your playthrough?

   _____

2) Did you find the game fun?
   - ☐ Yes
   - ☐ No
3) How easy do you think the game was to play?
   - ☐ Not at all
   - ☐ Not particularly easy
   - ☐ Somewhat Easy
   - ☐ Fairly Easy
   - ☐ Very Easy
4) If you were to rate how engaging the game was, what would you rate it on a scale of 1 to 10, where 1 being not engaging and 10 being very engaging?

   _____

5) Do you think the game was easy to navigate?
   - ☐ Yes
   - ☐ No
6) What was your favorite part, if any, of the game?

   _____

   _____

7) Do you have any suggestions for the game?

   _____

   _____

   _____

Section 2

1) Please see the following rule:

iptables -A INPUT -s 192.154.12.34 -p TCP –dport 80 -j DROP

Briefly describe what you think this rule is doing.

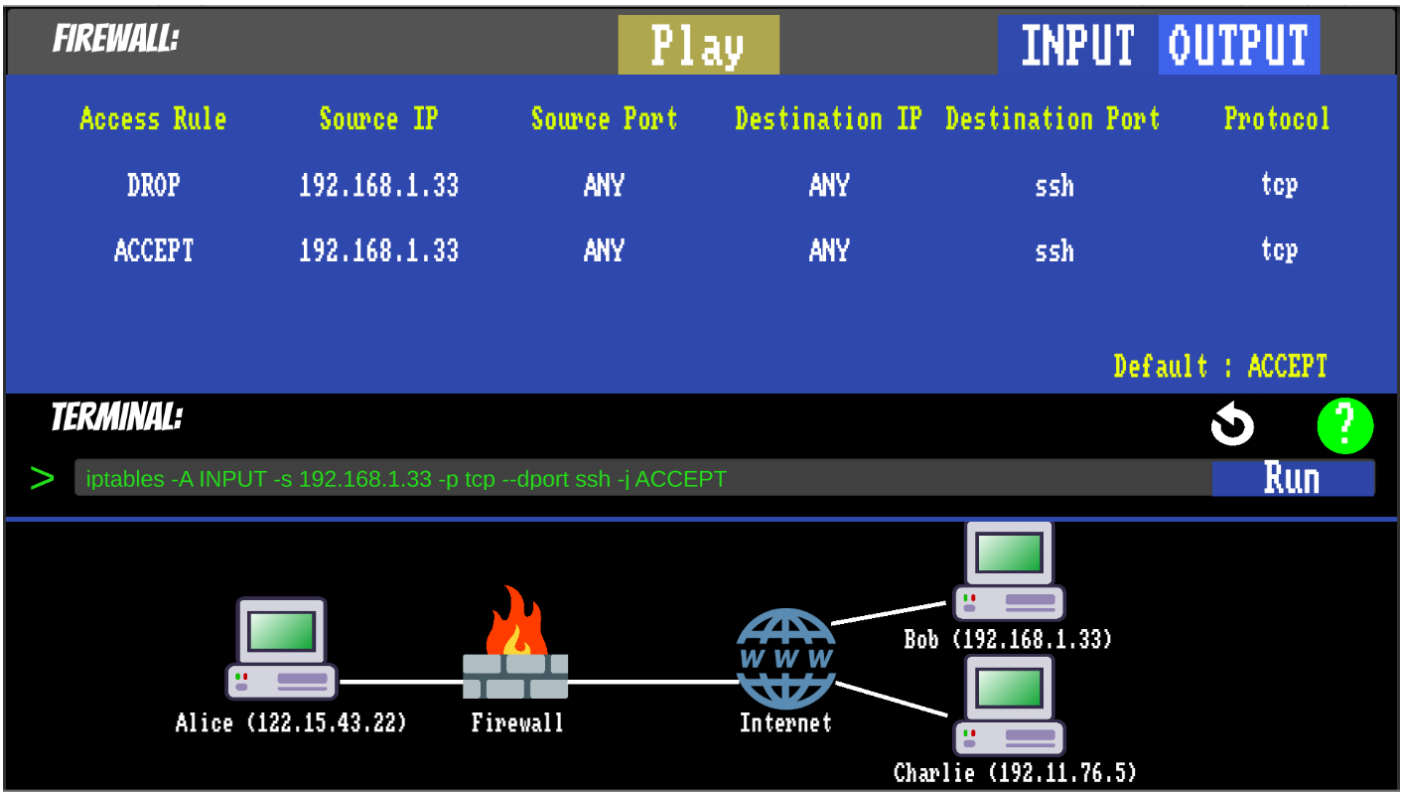
_____

_____

_____

*For the following questions, if you feel you do not have enough information in the question, please tick 'Cannot tell from the information given'

2) Consider the INPUT firewall chain as shown:



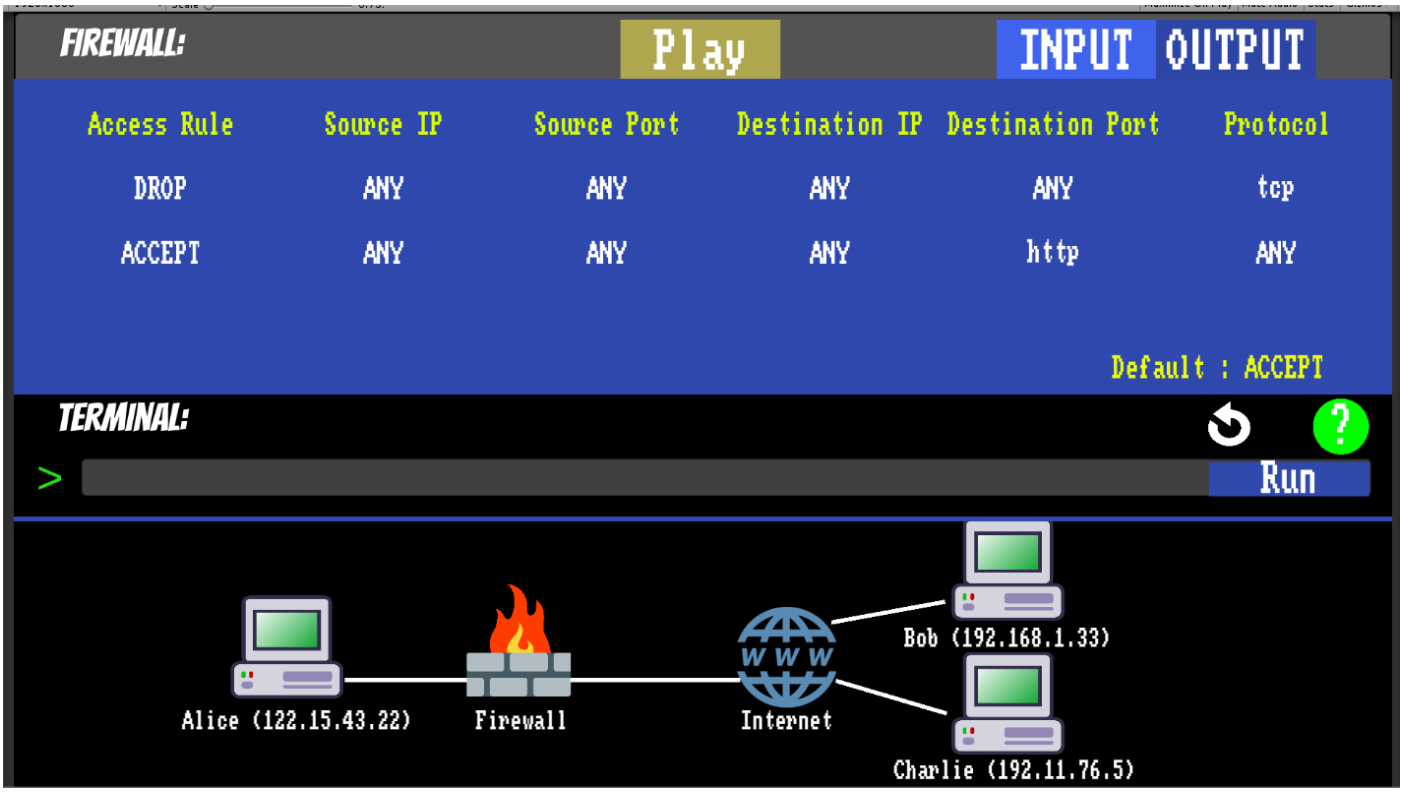
a) Do you think Bob would be able to SSH into Alice?

- ☐ Yes
- ☐ No
- ☐ Cannot tell from the information given

b) Do you think Charlie would be able to SSH into Alice?

- ☐ Yes
- ☐ No
- ☐ Cannot say

4) Consider the OUTPUT firewall chain:



a) If Bob was hosting a web server on the default port, do you think Alice could access it?

- ☐ Yes
- ☐ No
- ☐ Cannot say

b) Do you think Alice could SSH into Charlie's computer?

- ☐ Yes
- ☐ No
- ☐ Cannot say

Appendix F

Think-Aloud Script

Hello my name is Rusab. Today we will be playing a game about firewall administration to give you a good understanding of the way firewalls work and how they the rules can be configured using a Linux command line utility called 'iptables'. Try your best to solve all the levels, but your participation today is purely voluntary, you may stop at any time. The purpose of this game is to teach you how to set up your own firewall rules. Please remember we are testing this games capability to teach this, not your own.

Now I'm going to ask that you fill in this pre-game questionnaire. Please answer them as honestly as possible, as this is for the sake of research.

[Give them questionnaire and wait for them to finish it]

In this observation, we are interested in what you think about as you perform the tasks we are asking you to do. In order to do this, I am going to ask you to talk aloud as you work on the task. What I mean by "talk aloud" is that I want you to tell me everything you are thinking from the first time you see the statement of the task till you finish the task. I would like you to talk aloud constantly from the time I give you the task till you have completed it. I do not want you to try and plan out what you say or try to explain to me what you are saying. Just act as if you were alone, speaking to yourself. It is most important that you keep talking. If you are silent for a long period of time, I will ask you to talk. Do you understand what I want you to do?

Wait for a response

Good. Now we will begin with some practice problems. First, I will demonstrate by talking aloud while I solve a simple problem: "How many windows are there in my apartment?"

[Demonstrate thinking aloud]

Wait for response

Now it is your turn. Please talk aloud as you multiply 12 * 8.

[Let them finish]

Good. Now, those problems were solved all in our heads. However, when you are working on the computer you will also be looking for things and seeing things that catch your attention. These things that you are searching for and things that you see are as important for our observation as thoughts you are thinking from memory. So please verbalize these too. As you are playing the levels, I won't be able to answer any questions. But if you do have questions, go ahead and ask them anyway so I can learn more about what kinds of questions the game brings up that it hasn't explained. I will answer any questions after the session. Also, if you forget to talk aloud, I'll say, "please keep talking." Do you have any questions about the talk aloud?

[See if they ask a question]

Now, please proceed to play the game, and remember to read everything carefully.

[After game]

I hope you enjoyed the game. Please fill in this post-game questionnaire. Please answer them as honestly as possible, as this is for the sake of research.

Appendix G

SUS Survey

Feedback Survey

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

What is your gender?

☐ Male
☐ Female
☐ Prefer not to answer
☐ Other .....

What year are you?

☐ UG1
☐ UG2
☐ UG3
☐ UG4 / MInf
☐ MSc
☐ Other .....

Bibliography

[1] Mohammad Imran, Abdulrahman A. Alghamdi, and Bilal Ahmad. Role of firewall technology in network security. *International Journal of Innovation & advancement in Computer Science*, pages 3–6, 12 2015.

[2] Techopedia. Packet definition. https://www.techopedia.com/definition/5380/packet.

[3] Michael Shinn Scott Shinn. Introduction to troubleshooting linux firewalls. 01 2005.

[4] What is OSI model (Open Systems Interconnection)? https://searchnetworking.techtarget.com/definition/OSI.

[5] Dan Strom. The packet filter: A basic network security tool. https://cyber-defense.sans.org/resources/papers/gsec/packet-filter-basic-network-security-tool-100197.

[6] Habtamu Abie. An overview of firewall technologies. 12 2000.

[7] Avishai Wool. A quantitative study of firewall configuration errors. June 2004.

[8] Cornelius Diekmann, Lars Hupel, Julius Michaelis, Maximilian Haslbeck, and Georg Carle. Verified iptables firewall analysis and verification. *Journal of Automated Reasoning*, 61(1):191–242, Jun 2018.

[9] Wikipedia. iptables. https://en.wikipedia.org/wiki/Iptables, [AccessedJanuary25,2019].

[10] Prithviraj S. iptables tutorial. https://www.hostinger.co.uk/tutorials/iptables-tutorial#gref,[AccessedOctober20th,2017].

[11] Thawatchai Chomsiri, Xiangjian He, and Priyadarsi Nanda. Limitation of listed-rule firewall and the design of tree-rule firewall. pages 275–287, 2012.

[12] Alex Cruz Farmer. Firewall rules - priority and ordering. https://blog.cloudflare.com/firewall-rules-priority-and-ordering/,[AccessedDecember21th,2018].

[13] Marcus J. Ranum. Thinking about firewalls. http://alex.gray-world.net/es/papers/thinking.pdf.

[14] Maria Virvou, George Katsionis, and Konstantinos Manos. Combining software games with education: Evaluation of its educational effectiveness. *Educational Technology & Society*, 8:54–65, 04 2005.

[15] Alice Mitchell and Carol Savill-Smith. The use of computer and video games for learning: A review of the literature. 2004.

[16] Patrice Bouvier, Elise Lavou, and Karim Sehaba. Defining engagement and characterizing engaged-behaviors in digital gaming. *Simulation &amp Gaming*, 45:491–507, 11 2014.

[17] Alliance Media Holdings. Zachtronics. http://www.zachtronics.com/.

[18] Zachtronics. Exapunks. https://en.wikipedia.org/wiki/Exapunks.

[19] Opcode. Opcode. https://en.wikipedia.org/wiki/Opcode.

[20] Zachtronics. Tis-100. https://en.wikipedia.org/wiki/TIS-100.

[21] Zachtronics. Tis-100 steam. https://store.steampowered.com/app/370360/TIS100/?curatorclanid=32946839.

[22] Zachtronics. Texapunks steam. https://store.steampowered.com/app/716490/EXAPUNKS/.

[23] Sibylle Sehl and Kami Vaniea. Permission impossible: Teaching firewall configuration in a game environment. 2018.

[24] Charles. Blue team. https://groups.inf.ed.ac.uk/tulips/projects/1617/CharlesFirewallGameWebsite/Website/.

[25] Adobe. Adobe photoshop. https://www.adobe.com/uk/products/photoshop.html.

[26] Blender. https://www.blender.org/.

[27] Git. 2009. https://git-scm.com/.

[28] Microsoft. Github. https://github.com/.

[29] Opengameart. https://opengameart.org/.

[30] University of Edinburgh. Computer security. https://www.inf.ed.ac.uk/teaching/courses/cs/.

[31] Google. Google forms. https://www.google.co.uk/forms/about/.

[32] Unity profiler. https://docs.unity3d.com/Manual/Profiler.html.

[33] Intel core i5-6200u processor. https://ark.intel.com/content/www/us/en/ark/products/88193/intel-core-i5-6200u-processor-3m-cache-up-to-2-80-ghz.html.

[34] Elizabeth Charters. The use of think-aloud methods in qualitative research, an introduction to think-aloud methods. https://journals.library.brocku.ca/brocked/index.php/home/article/view/38/38.

[35] Usability Gov. System usability scale. `https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html`.

[36] Shahedul Huq Khandkar. Open coding. *University of Calgary*, 23, 2009.

[37] Nathan Thomas. SUS Score Calculation Technique. https://usabilitygeek.com/how-to-use-the-system-usability- scale-sus-to-evaluate-the-usability-of-your-website/.

[38] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.