CyberBlock Game

CyberBlock Team
Nelson Lau
Zach Kelly
Ridhisha Dangol
Xin (Jason) Li

Software Requirements Specification

Version: 2 **Date: 10/28/2021**

Table of Contents

1. Introduction **4**

1.1 Purpose *4*

1.2 Scope *4*

1.3 Definitions, Acronyms, and Abbreviations. *4*

1.4 References *4*

1.5 Overview *5*

2. The Overall Description **5**

2.1 Product Perspective *5*
2.1.1 System Interfaces 5
2.1.2 Interfaces 5
2.1.3 Hardware Interfaces 6
2.1.4 Software Interfaces 6
2.1.5 Communications Interfaces 6
2.1.6 Memory Constraints 6
2.1.7 Operations 6
2.1.8 Site Adaptation Requirements 7

2.2 Product Functions *7*

2.3 User Characteristics *7*

2.4 Constraints *7*

2.5 Assumptions and Dependencies *7*

2.6 Apportioning of Requirements. *8*

3. Specific Requirements **8**

3.1 External Interfaces *8*
3.1.1 User Interfaces 8
3.1.2 Hardware Interfaces 8
3.1.3 Communications Interfaces 8

3.2 System Features *9*
3.2.1 Create a Game Lobby 9
3.2.1.1 Introduction/Purpose of Features 9
3.2.1.2 Stimulus/Response Sequence 9
3.2.1.3 Associated Functional Requirements 9
3.2.1.3.1 Functional Requirement 1 9
3.2.1.3.2 Functional Requirement 2 9
3.2.2 Players Joining the Game 10
3.2.2.1 Introduction/Purpose of Features 10
3.2.2.2 Stimulus/Response Sequence 10
3.2.2.3 Associated Functional Requirements 10
3.2.2.3.1 Functional Requirement 1 10
3.2.3 Trivia Game 10
3.2.3.1 Introduction/Purpose of Features 10
3.2.3.2 Stimulus/Response Sequence 10
3.2.3.3 Associated Functional Requirements 10
3.2.3.3.1 Functional Requirement 1 10

3.2.4 Attacks are Determined 11
3.2.4.1 Introduction/Purpose of Features 11
3.2.4.2 Stimulus/Response Sequence 11
3.2.4.3 Associated Functional Requirements 11
3.2.4.3.1 Functional Requirement 1 11
3.2.5 Defending Against Attacks 11
3.2.5.1 Introduction/Purpose of Features 11
3.2.5.2 Stimulus/Response Sequence 11
3.2.5.3 Associated Functional Requirements 12
3.2.5.3.1 Functional Requirement 1 12
3.2.5.3.2 Functional Requirement 2 12
3.2.6 Points are Awarded 12
3.2.6.1 Introduction/Purpose of Features 12
3.2.6.2 Stimulus/Response Sequence 13
3.2.6.3 Associated Functional Requirements 13
3.2.6.3.1 Functional Requirement 1 13
3.2.6.3.2 Functional Requirement 2 13
3.2.7 Scoreboard 13
3.2.7.1 Introduction/Purpose of Features 13
3.2.7.2 Stimulus/Response Sequence 13
3.2.7.3 Associated Functional Requirements 13
3.2.7.3.1 Functional Requirement 1 13

3.3 Performance Requirements 13

3.4 Design Constraints 14
3.4.1 Standards Compliance 14

3.5 Logical Database Requirements 14

3.6 Software System Attributes 14
3.6.1 Reliability 14
3.6.2 Availability 14
3.6.3 Security 14
3.6.4 Maintainability 14
3.6.5 Portability 15

3.7 Additional Comments 15

4. Change Management Proces 15

5. Document Approvals 15

6. Supporting Information 15

6.1 Appendix A: Mitre Matrix 15

6.2 Appendix B: Cheat Sheet 16

1. Introduction

1.1 Purpose

The CyberBlock Game Software Requirements Specification defines the necessary interfaces, functions, and requirements specified for the project. The intended audience of the CyberBlock Game Software Requirements Specification are Medtronic employees who go through cybersecurity training.

1.2 Scope

The purpose of the CyberBlock game is to engage students and workers involved in the medical device security field by having users purchase upgrades and defenses in a game environment that could potentially stop cyber-attacks similar to situations that they would face in the real world. With a web-based version of the game, participants will learn how to counter cyber-attacks using defenses that could be implemented into medical devices and from the interactions, we hope to provide an enjoyable experience for the players.

1.3 Definitions, Acronyms, and Abbreviations.

Lobby-based Game	Staging area where players join before the actual game begins.
Game Server	Transmits data to allow users to maintain and display an accurate version of the game. User inputs are received and processed.
AD/LDAP	Active Directory/Lightweight Directory Access Protocol

1.4 References

"Mitre ATT&CK." *MITRE ATT&CK*, The MITRE Corporation, 2021, https://attack.mitre.org/.

"D3fend Matrix: Mitre D3FEND." *D3FEND Matrix | MITRE D3FEND*, The MITRE Corporation, 2021, https://d3fend.mitre.org/.

"General Cyber Security Taxonomy", University of Kent, 2021, https://cyber.kent.ac.uk/research/cyber_taxonomy/app/taxonomy/visualization?t=space_tree&n=Cyber%2BSecurity&st=true&nid=39.

"Common Attack Pattern Enumeration and Classification." *CAPEC*, The MITRE Corporation, 25 Feb. 2021, https://capec.mitre.org/.

Tram-Anh Cai. "CyberBlock Components", Medtronic, 09/09/2021

1.5 Overview

Medical device manufacturers, healthcare providers, and regulators are increasingly facing issues with the security of medical devices. While it can be simple to make medical devices more secure, the trade-off is the loss of efficient devices. The project can be used in the classroom as well as remote to promote the game as a training exercise to learn how to spend resources on defending against cyber-attacks. The computer-based game will have features to allow teams to purchase security controls for their devices using limited resources. Hosts will create game lobbies that users will be able to join using a game code. The users will then face off with cyber-attacks against their devices which are determined based on a roll of a die. If the team can fend off the attack, whether from eliminating or remediating the attack, they will be awarded points. At the end of the game, the team will be able to view the number of points received through a scoreboard. The product is planned to be released as a browser-based game which can be used in two browsers (Google Chrome and Internet Explorer).

2. The Overall Description

2.1 Product Perspective

The CyberBlock game is independent and self-contained. The game will allow users to join through a web browser in order to access the game. Cyber security instructors will be able to create game lobbies where users can join the game and play along. The game's purpose is to train students or employees how to manage and prepare for cyber security risks via defenses and learn which defenses work well against which attacks. There is no immediately similar product on the market.

2.1.1 System Interfaces

The system will need to interface with the database server via MySQL API calls to perform SQL queries.

2.1.2 Interfaces

The product will be accessed by users via a web-based GUI. The specific GUI requirements are:

- Players will be able to enter a code in a text box to join a specific lobby.
- Players will have an in-game leaderboard to display team points and current round.
- Players are able to see their team's current cards.
- Players are able to see the current attack category for a round.
- Players are able to see the cards that other teams have selected during each round.
- Players are able to answer questions in a trivia round before game starts to earn extra money for buying defenses.
- Instructors will have a menu to create a new lobby and change game aspects

  - Game time
  - Round count
  - Team count
- Instructors will be able to move players to and from different teams.
- Instructors will be able to see the same leaderboard as the players.
- Instructors will be able to randomly roll for an attack category and select an attack from given category.
- Administrators will be able to add, delete, or modify attacks and defenses through an administrator GUI.

2.1.3 Hardware Interfaces

The users will interact with the software using a keyboard and a mouse/trackpad through their web browser by navigating to the game's website. Devices will require a network interface card for network connectivity and must be able to use either ethernet or WiFi. Devices will need a web-browser built into the computer capable of HTML5.

2.1.4 Software Interfaces

The software will read data to and from a MySQL database for the storage of attack and defense data.

2.1.5 Communications Interfaces

The users will connect to the game server via a web interface through their web browser. It will be using socket.io to manage client-server communication.

The game will be compatible with Chromium-based web browsers and Internet Explorer.

The game server will be communicating with the database with SQL queries.

2.1.6 Memory Constraints

There are no specific memory constraints.

2.1.7 Operations

When the game server determines that it is a team's turn to play, the players will be able to interact with the game using their keyboard and mouse. When the game server determines that it is not the team's turn to play, such as periods where it is the instructor's turn, the players will be able to watch the game but will not be able to interact with the game.

All data processing, such as reading from the database, will be handled by the game server.

2.1.8 Site Adaptation Requirements

As the product is going to be used from a web interface, the client device must have internet access and a HTML5 compatible web browser.

2.2 Product Functions

The product is designed to imitate the card-based game known as CyberBlock. Players will use their web browser to navigate to the game website and will interact with the web GUI to join game lobbies. Instructors will be able to create game lobbies and will have a randomly generated code to join the lobbies. The GUI will contain general information about the current game via the leaderboard. Players will use their mouse to interact with cards and select cards to play.

2.3 User Characteristics

The users for the game are divided into two categories: beginner, intermediate and advanced players. The beginner's category will have reduced card sets based on the attack score below 5. The difficult mode will have all the cards set. The beginner user's interface will have the least number of cards sets and the advanced level users will have the greatest number of cards. The attacks will have difficulty as attribute to determine whether the attack is beginner, intermediate or advanced level attack.

2.4 Constraints

The few major constraints of the game are as follows:

- The game needs to be supported on Windows, Linux, and IOS and run as a website needs to run on the following browsers: Google Chrome and Internet Explorer.
- The user should not have to install any software to play the game.
- The game will run on Google and Internet Explorer.
- The game will have an instructor mode to host the game and a player mode to play.
- The web application will run on Heroku and accessible to anyone.
- The data for the game is obtained from the Mitre attack and defend matrix.

2.5 Assumptions and Dependencies

The following as the assumptions made:

- The data for the game is obtained from the Mitre Attack and Defend matrix (Appendix A).
- The user is familiar with using internet browsers and computer peripherals.
- The game will be a web application and the internet will be required to play.
- We will have the necessary development team of 4 software developers to finish the project given the time constraint.
- Any required maintenance will mainly be done during non-office hours and all the devices needed for the server are in good condition.

The following are the dependencies:

- The server will have to be online 24/7.
- A prototype of the project will need to be completed by the first week of December.
- In case of a server or a database crash, the site will be down.

2.6 Apportioning of Requirements.

The following are priorities that need to be implemented in the prototypes:

- The game needs to implement all the attacks and defenses included in the cards.
- The game implements all the attacks and defenses included in the cards.
- The web app needs to run on the two main browsers Chrome and Internet explorer.
- There should be three types of game difficulty: beginner, intermediate and advanced.
- Medtronic would preserve full control over the game servers during the game development.
- The game will have a chat box feature to allow players in a team to communicate.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

User Interfaces: For the game section, The Graphical user interface will be developed using HTML, CSS styling.

There will be two user interfaces an instructor page and player page.

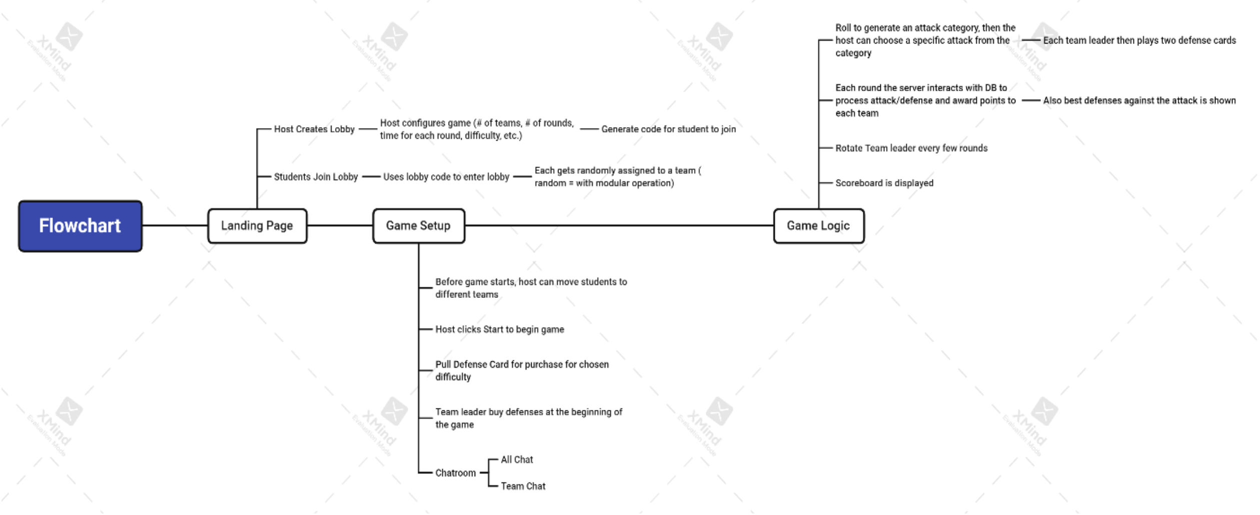
3.1.2 Hardware Interfaces

Hardware will comprise of a mouse/trackpad and keyboard through the web interface. Devices will require a network interface card for network connectivity and must be able to use either ethernet or Wi-Fi. The software will include mouse functionality like click while the keyboard will be used for communication purposes.

3.1.3 Communications Interfaces

The product will connect to a local network connection over Wi-Fi or Ethernet to access the hosted site on the internet.

3.2 System Features


Flowchart
Landing Page
Host Creates Lobby
Host configures game (# of teams, # of rounds, time for each round, difficulty, etc.)
Generate code for student to join
Students Join Lobby
Uses lobby code to enter lobby
Each gets randomly assigned to a team (random = with modular operation)
Game Setup
Before game starts, host can move students to different teams
Host clicks Start to begin game
Pull Defense Card for purchase for chosen difficulty
Team leader buy defenses at the beginning of the game
Chatroom
All Chat
Team Chat
Game Logic
Roll to generate an attack category, then the host can choose a specific attack from the category
Each team leader then plays two defense cards
Each round the server interacts with DB to process attack/defense and award points to each team
Also best defenses against the attack is shown
Rotate Team leader every few rounds
Scoreboard is displayed

3.2.1 Create a Game Lobby

3.2.1.1 Introduction/Purpose of Features

The host must create a lobby in which players are able to join. The host will be able to choose the difficult mode (either beginner, intermediate or advanced) of the lobby and set the number of rounds for the game.

3.2.1.2 Stimulus/Response Sequence

Users will navigate through the lobby creation using mouse and keyboard stimuli. The program will set a game based on the difficulty mode and the number of rounds per game based on the instructor's input and implement those changes to the game.

3.2.1.3 Associated Functional Requirements

3.2.1.3.1 Functional Requirement 1

The host should be able to enter the difficulty mode and the number of rounds for the game.

3.2.1.3.2 Functional Requirement 2

Lobby codes should be generated for the game in order to allow players to join.

3.2.2 Players Joining the Game

3.2.2.1 Introduction/Purpose of Features

When the game code is generated by the host, all the players will be able to enter that specific game code and join the lobby created by the instructor.

3.2.2.2 Stimulus/Response Sequence

Players will be able to join lobbies by clicking the ‘join lobby’ button. The program will determine what lobby is associated with that specific game code and allow the player to join.

3.2.2.3 Associated Functional Requirements

3.2.2.3.1 Functional Requirement 1

Users can enter the game code to connect them to the game. The player will randomly be assigned to a team consisting of 3-5 players depending on the total number of players. A single player will be randomly chosen to lead the team. The time period for the assigned leadership will be determined by getting total players in a team by the total number of rounds. For example, if a team of 4 players play a total of 8 rounds, then an individual will be the leader for two rounds such that each player will have an opportunity to have leadership ability.

The leadership abilities include submitting the answer button after the team has discussed their opinions via a chat box.

3.2.3 Trivia Game

3.2.3.1 Introduction/Purpose of Features

The players will have a trivia question with answers as options at the beginning as an opportunity to earn more money. The user can select an option and obtain and award points for game play.

3.2.3.2 Stimulus/Response Sequence

When the player inputs a mouse click over one of the cards in their hand, a database query will determine if the card selected is defense that is applicable to the attack.

3.2.3.3 Associated Functional Requirements

3.2.3.3.1 Functional Requirement 1

The user screen will have a trivia question which will be a multiple-choice question. The player will have to select an answer in order to proceed. The correct answer will result in an award of $5 and if the answer is incorrect the player will not receive any points.

3.2.4 Attacks are Determined

3.2.4.1 Introduction/Purpose of Features

At the beginning of each round, an attack is determined by rolling two dices. The higher probability numbers will be given more common real-world attacks.

3.2.4.2 Stimulus/Response Sequence

The first attack will be rolled when the instructor presses the button called 'begin game' button. All rounds after the first will not require a trigger via a button but will be done using set timers for an action.

3.2.4.3 Associated Functional Requirements

3.2.4.3.1 Functional Requirement 1

The program will randomly generate a number as a simulation of rolling two dices, which represents an attack category.

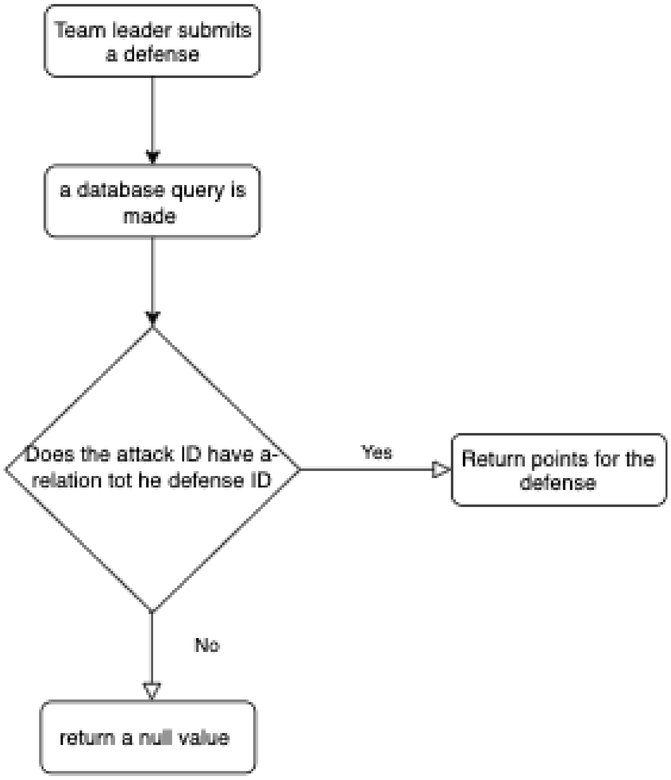
3.2.5 Defending Against Attacks

3.2.5.1 Introduction/Purpose of Features

The players discuss among themselves using the chat box and the team leader will be the only one with the ability to choose one or two cards, depending on the round. In the first round, the team leader can only use one defense card but after the first round the team can use up to 2 defense cards. The server will determine if the defense is suitable against the attack by checking the database.

3.2.5.2 Stimulus/Response Sequence

When the user inputs a mouse click over one of the cards in their hand, a database query will determine if the card selected is defense that is applicable to the attack.


Team leader submits a defense
a database query is made
Does the attack ID have a relation tot he defense ID
Yes
Return points for the defense
No
return a null value

3.2.5.3 Associated Functional Requirements

3.2.5.3.1 Functional Requirement 1

Users should be able to click on a card to use its defense for the round.

3.2.5.3.2 Functional Requirement 2

The program will need to determine if the defending cards are effective against the current attack. It is done using a database query to see if the defense card has the Attack ID associated with the defense being played and if so, a point associated with it.

3.2.6 Points are Awarded

3.2.6.1 Introduction/Purpose of Features

When defenses are successful, points will be awarded to the teams based on the effectiveness of the defense against the attack. If a team is unsuccessful in defending against an attack, the teams will not be awarded points.

3.2.6.2 Stimulus/Response Sequence

The points are gained when the user selects a correct defense against an attack. The points will be stored temporarily in the local storage of the webpage and will have updates as the game proceeds.

3.2.6.3 Associated Functional Requirements

3.2.6.3.1 Functional Requirement 1

A function will determine the value of points that are awarded to teams based on their defense's effectiveness against attacks.

3.2.6.3.2 Functional Requirement 2

Points will need to be stored in the local storage of the webpage.

3.2.7 Scoreboard

3.2.7.1 Introduction/Purpose of Features

There will be a scoreboard that will display the total amount of points that each team earned during the game.

3.2.7.2 Stimulus/Response Sequence

The total amount of points that each team earned during the game will display.

3.2.7.3 Associated Functional Requirements

3.2.7.3.1 Functional Requirement 1

The total points earned by teams will be shown as part of the user interface.

3.3 Performance Requirements

The following are the performance requirements.

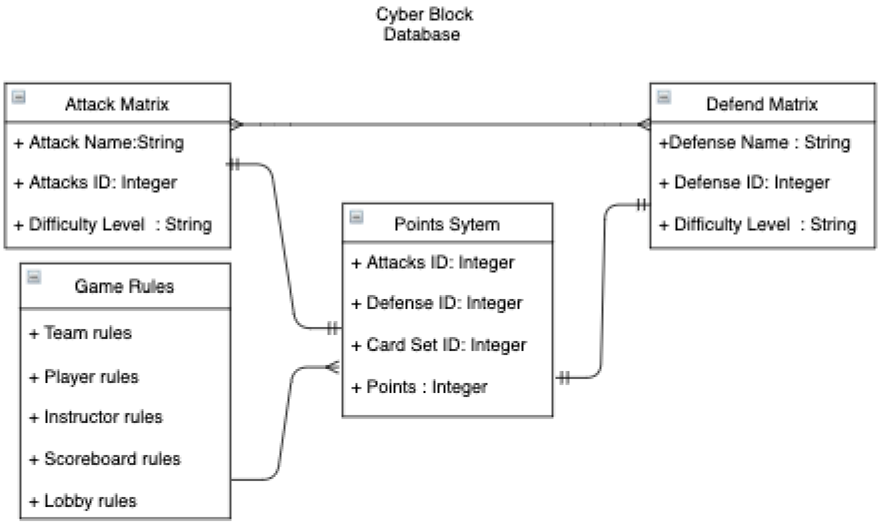
- The web app needs to handle a minimum of 20 teams with an average team size of 3-5 players during peak hours (9am-5pm).
- The average time to store information in a database should be at the most 2 seconds.
- The application response time shouldn’t exceed a window of 5 seconds for user interactivity accounting for any delays.
- The app needs to be scalable for larger screens and smaller screens with a minimum of an iPad screen.
- The product needs to function on all cross platforms of only Windows and IOS.
- All GUI elements with interactive features should respond within 0.3 seconds.
- All form-based transactions (user events like clicking a button) should be completed within 5 seconds with basic error checking and validations.

3.4 Design Constraints

3.4.1 Standards Compliance

Google and Internet Explorer: Since the app is web-based, it needs to be compatible across chrome and Internet Explorer.

3.5 Logical Database Requirements



3.6 Software System Attributes

3.6.1 Reliability

Under normal usage conditions, the application should run for a whole game session without failing.

3.6.2 Availability

The application is available all the time as a web application.

3.6.3 Security

Since the application does not deal with any confidential data, security is not a major concern.

3.6.4 Maintainability

The administrator can add and delete attacks or defenses to the database.

3.6.5 Portability

The application will run on Chrome and Internet Explorer.

3.7 Additional Comments

There are not additional comments for version 2 of the Software Requirements Specification.

4. Change Management Proces

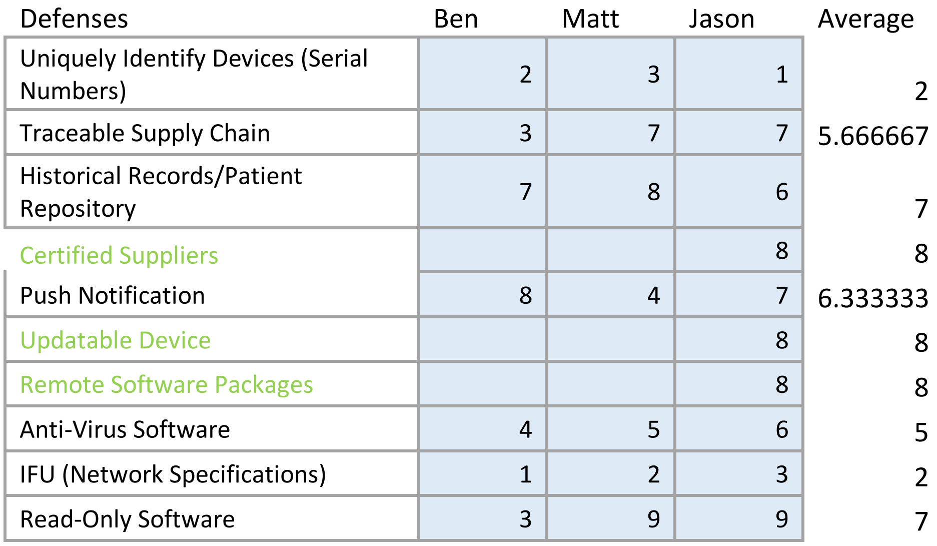
All changes made to the document will need to be evaluated by both team members and clients. Once all parties come to a unanimous decision about the changes, the updates will be implemented into the document. Previous versions will be saved as separate documentation while new versions will act as the logs to the changes.

5. Document Approvals

Name	Signature	Date

6. Supporting Information

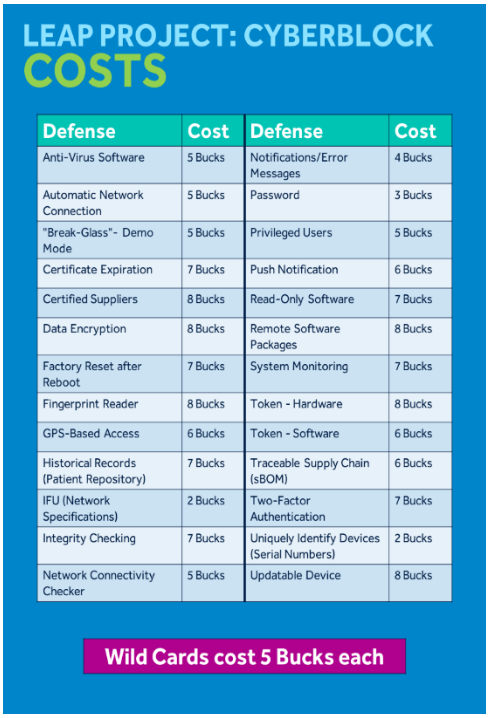
6.1 Appendix A: Mitre Matrix



Defenses	Ben	Matt	Jason	Average
Uniquely Identify Devices (Serial Numbers)	2	3	1	2
Traceable Supply Chain	3	7	7	5.666667
Historical Records/Patient Repository	7	8	6	7
Certified Suppliers			8	8
Push Notification	8	4	7	6.333333
Updatable Device			8	8
Remote Software Packages			8	8
Anti-Virus Software	4	5	6	5
IFU (Network Specifications)	1	2	3	2
Read-Only Software	3	9	9	7

Factory Reset after Reboot	7	7	6	6.666667
Privileged Users	7	5	3	5
System Monitoring	7	8	7	7.333333
Integrity Checking	7	8	7	7.333333
Data Encryption	10	6	7	7.666667
Password	2	2	4	2.666667
Token - Hardware	8	8	7	7.666667
Token - Software	5	7	6	6
Fingerprint Reader	7	9	7	7.666667
GPS-Based Access	4	8	6	6
Two-Factor Authentication	7	8	7	7.333333
"Break-Glass"- Demo Mode	3	6	7	5.333333
Certificate Expiration			7	7
Notifications/Error Messages	2	4	5	3.666667
Automatic Network Connection	3	7	4	4.666667
Network Connectivity Checker	4	8	4	5.333333

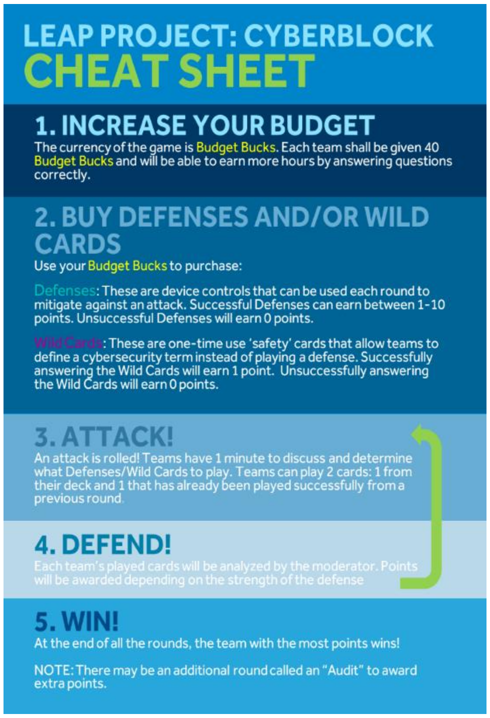
6.2 Appendix B: Cheat Sheet



LEAP PROJECT: CYBERBLOCK
COSTS

Defense	Cost	Defense	Cost
Anti-Virus Software	5 Bucks	Notifications/Error Messages	4 Bucks
Automatic Network Connection	5 Bucks	Password	3 Bucks
"Break-Glass"- Demo Mode	5 Bucks	Privileged Users	5 Bucks
Certificate Expiration	7 Bucks	Push Notification	6 Bucks
Certified Suppliers	8 Bucks	Read-Only Software	7 Bucks
Data Encryption	8 Bucks	Remote Software Packages	8 Bucks
Factory Reset after Reboot	7 Bucks	System Monitoring	7 Bucks
Fingerprint Reader	8 Bucks	Token - Hardware	8 Bucks
GPS-Based Access	6 Bucks	Token - Software	6 Bucks
Historical Records (Patient Repository)	7 Bucks	Traceable Supply Chain (sBOM)	6 Bucks
IFU (Network Specifications)	2 Bucks	Two-Factor Authentication	7 Bucks
Integrity Checking	7 Bucks	Uniquely Identify Devices (Serial Numbers)	2 Bucks
Network Connectivity Checker	5 Bucks	Updatable Device	8 Bucks

Wild Cards cost 5 Bucks each



LEAP PROJECT: CYBERBLOCK
CHEAT SHEET

1. INCREASE YOUR BUDGET

The currency of the game is Budget Bucks. Each team shall be given 40 Budget Bucks and will be able to earn more hours by answering questions correctly.

2. BUY DEFENSES AND/OR WILD CARDS

Use your Budget Bucks to purchase:

Defenses: These are device controls that can be used each round to mitigate against an attack. Successful Defenses can earn between 1-10 points. Unsuccessful Defenses will earn 0 points.

Wild Cards: These are one-time use 'safety' cards that allow teams to define a cybersecurity term instead of playing a defense. Successfully answering the Wild Cards will earn 1 point. Unsuccessfully answering the Wild Cards will earn 0 points.

3. ATTACK!

An attack is rolled! Teams have 1 minute to discuss and determine what Defenses/Wild Cards to play. Teams can play 2 cards: 1 from their deck and 1 that has already been played successfully from a previous round.

4. DEFEND!

Each team's played cards will be analyzed by the moderator. Points will be awarded depending on the strength of the defense

5. WIN!

At the end of all the rounds, the team with the most points wins!

NOTE: There may be an additional round called an "Audit" to award extra points.