IRM Project Requirements

Miles DeBoer

May 21st, 2024

Table of Contents

1. Introduction

1.1 Scope

This document is intended to be read by people involved in the development of the game.

2. Development System

2.1 Software

The game will be developed using the Unity Engine 2022.3.21f1 and Visual Studio Code.

2.2 Languages

The game will be developed using C#.

3. Overview

The goal of playing the serious game will be to be educated on different types of cyber threats and the process involved in preventing and handling them once they occur.
The goal for the player will be to gather information and the funds of other players by sending malware and preventing their information and funds from being exposed/stolen by others.
The player wins once the funds of other players run out or an overall goal is achieved.

3.1 Story

3.1.1 Plot

The player will commit corporate espionage in order to make sure that their business is the last one remaining.

3.1.2 Character

The player will play the head of the department of cyber security for a business they create.

3.1.3 Conflict

Each player is competing by attacking each other’s systems.

3.1.4 Theme

The theme of the game is Cyber Security.

3.1.5 Setting

The game will be set in the present-day world.

3.2 Gameplay Mechanics

The four main mechanics that the player should be able to do is to create custom malware, customize the defenses of their data centers, plan the attack of another player, and progress through the overall goal. Data centers produce money and resources for the player to build and improve things such as malware and data center upgrades.

3.4 Players

The PC game will be playable by multiple users on a single device or possibly across a P2P network on multiple devices. Users will also have the option to play a one-player version on a single device.

3.6 Objective

A player wins the game when they develop a fully autonomous AI system (or other arbitrary overall goal) or is the last player remaining. Similarly, a player loses when another player successfully completes this arbitrary goal or when they run out of funds.
The progress of the overall goal will be presented through a tree of arbitrary milestones to make the progress seem less linear and allow the player to steal the progress of other players

4. Gameplay

4.1 Offense

The player should be able to go through the seven stages of the cyber kill chain (reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), and actions on objectives).

- **Reconnaissance**: The player should be able to gather information on the system of the target through network scanning or database searches.
- **Weaponization**: The player should be able to create custom malware deployment such as an infected pdf or a physical device.
- **Delivery**: The player should be able to deploy their malware through phishing campaigns, manual deployment or an existing backdoor.
- **Exploitation**: The player should be able to customize the target exploit. (lure user to executing file or staging malware)
- **Installation**: The player should determine what happened when the malware is installed. (backdoor setup, etc.)
- **Command and Control (C2)**: The user should be able to tell the malware what to do once it is already installed. (what to scan, what to take, etc.)
- **Actions on Objectives**: The player should be able to determine the final goal of the malware.

4.2 Standard Attack Process

When a player launches an attack on another player, the following process will occur

- The game manager will take in the attributes of both the attack and the defense
    - Attack Attributes
        - Stealth: Decreases the likelihood of being detected.
        - Size: Increases the likelihood of being detected.
        - Exploit: Decreases the likelihood of being detected if used.
        - Reconnaissance: Decreases the likelihood of being detected if the data center's defenses have not changed since the time of the reconnaissance.
    - Defense Attributes
        - IDS: Thresholds the overall stealth of the attack (combination of stealth and size) so that low stealth attacks notify the player who owns the data center.
        - IPS: Thresholds the overall stealth of the attack so that low stealth attacks are prevented from infecting the system. This threshold is
        - Firewall: Thresholds the size of attacks so that large size attacks cannot take place at the cost of the production of the data center. This threshold is customizable by the player where the lower the threshold, the lower the production.
        - Learning Algorithm: Increases the likelihood of the malware not being successful if malware with the same ID has infected this data center before. Less effective version if another data center of the owner has been infected by this malware.
- Using these attributes, the game manager will determine the chance of the attack being successful.
- If the attack is successful, the goal of the attack is achieved. In general, more ambitious attacks are more difficult to pull off.

4.3 Phishing Attack Process

During a player's turn, there will be a list of emails that the player can accept or decline found at any of that user's data centers. By pressing accept, they run the risk of letting malware into that data center, but if there is none, the production of that data center is boosted. Malicious emails contain small discrepancies compared to valid emails.

4.4 Defense

The player should be able to defend against the seven stages of the cyber kill chain (reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), and actions on objectives) by spending resources to build upgrades on a data center's defense.

- **Reconnaissance**: The player should be able to add measures to hide network structure.
- **Weaponization**: The player should be able to update the system to patch vulnerabilities.
- **Delivery**: The player should be able to set up email filtering and media scanning. Purchasing and managing a security system should also be available to the player.
- **Exploitation**: The player should be able to scan for vulnerabilities.

- **Installation**: The player should be able to create anti-malware.
- **Command and Control (C2)**: The player should be able to analyze network traffic.
- **Actions on Objectives**: The player should be able to set up data encryption.

Most of the features of data centers can also be improved to increase their effect including money production and the amount of resources.

4.5 Goal Completion

The player should be able to progress towards an arbitrary goal by using resources to complete a series of tasks on a tree until the tree is completed and the game is won.

5. Art and Design

The player will be operating a computer with limited function for the entirety of the game except when paused or in the main menu.

5.1 Taskbar

The taskbar will be made up of a series of icons corresponding to a different window.

5.2 Data Centers

The data center window will show a map of the data centers.
By pressing a data center owned by the player, it will pull up a window showing specification of that data center

5.2.1 Data Center Customization

This window will include a series of fields to improve the defense of the data center.

- Option to hide network structure (Button). Decreases success rate of recon
- Scan for vulnerabilities (Button)
- Patch vulnerabilities (Button)
- IDS (Button). Alerts player when malware stealth is below threshold.
- IPS which can only be selected if IDS is built (Button)
- A Firewall display will show a slider to determine the strength of the firewall.
- Data Encryption (Button). Decrease amount of data stolen.
- Data Loss Prevention (Button). Decrease the amount of money stolen.
- Email Filtering (Button)
- List of Emails that the user will have to accept or deny which will boost player production.
- List of Files that the user will have the option to delete with a confirmation prompt

5.3 Malware

The malware window will show a list of existing malware, a button to create new malware and a progress bar showing any work being done in malware development.

Pressing any malware on the list will bring up a separate menu for that specific piece of malware where the player can choose how to develop that malware. Pressing the create new malware button will do something similar, but the new menu will be empty.

5.3.1 Malware Customization

This window is made up of a series of fields to increase the effectiveness

- A series of buttons across the top for the player to determine the type of malware (virus, worm, etc.). The user must select only one of these options.
- A series of sliders for each attribute of the malware (speed, stealth, etc.)
- A series of buttons for each special feature to the malware which increase and decrease certain attributes of the malware. This increase and decrease is not displayed in the sliders of the attributes.
- A space where the user can adjust the amount of resources going into the production of this malware.
- Text displaying a time estimate as to when the creation of the malware will be completed. Higher resources corresponds to less time.
- A finish button

Hovering over any attribute or feature on this page should display a small amount of flavor text briefly explaining what that feature is in real life and its effect in the game.

5.4 Attacks

The attack window will show a list of existing attacks and a button to create a new attack. There will also be a progress bar showing any progress being done on attacks.

Pressing any of the existing attacks will bring up a separate menu for that specific attack where the player can choose the details of the attack. Pressing the new attack button will similarly create a menu that has nothing selected.

5.4.1 Attack Customization

This window is made up of a series of fields to increase the effectiveness

- A map of data centers that the player can select as an attack target.
- A selection menu of the malware you are using for the attack. Reconnaissance is also an option on this list.
- A list of options where the player can choose the goal of the malware (steal money, steal knowledge, etc.).
- A list of options where the player can choose the method of delivery for the attack.
- A button that allows you to select a vulnerability to exploit if one exists. Can only be used if the player previously created one.
- A space where the user can adjust the amount of resources going into the production of the attack.

- Text displaying a time estimate as to when the attack will be completed. Higher resources corresponds to less time.
- A finish button.

5.5 Goal Progress

The goal progress window will be a display of a progress tree showing the progress towards the overall goal and a progress bar showing any work being done on this tree (if any).
Each node of the tree should hold a small amount of flavor text explaining that stage.
The player can select an available node to start work on that task.

5.6 Pause Menu

The pause menu will include the following options.

- Unpause: A button that will unpause the game.
- Save: A button that will save the current game state.
- Load: A button that will pull up a sub menu where the player can select the game state that will be loaded
- Main Menu: A button that will exit to the main menu. This will prompt the user to confirm as this will not automatically save the game.
- Exit: A button that will exit the game entirely. This will prompt the user to confirm as this will not automatically save the game.

5.7 Main Menu

The main menu will include the following options.

- New Game: This button will bring up a separate window where the player can select game settings:
    - Number of Players
    - Number of Computers
- Load Game: This button will bring up a separate window where the player can select a game state that will be loaded.
- Exit: This button will exit the game.

6. Data Storage

6.1 Game

The following information is the data regarding game settings that is to be stored when the user saves the game.

- Num Players: int - the number of players in the game.
- Num Computers: int - the number of computer players in the game.

6.2 Player

The following information is the data regarding players that is to be stored when the user saves the game.

- Player ID: int - the identification number of the player
- Resources: int - the number of available resources for the player
- Money: int - the amount of money the player has

6.3 Data Centers

The following information is the data regarding data centers that is to be stored when the user saves the game.

- Id: int - the identification number of the data center
- Owner: int - the player number of the owner of the data center
- Production Rate: int - the rate at which money is produced.
- Money: int - the amount of money stored at this location.
- Malware: Malware - List of malware that this data center is infected with. Stored as list of numbers (malware IDs)
- Vulnerabilities: - List of player IDs representing which players have created some kind of vulnerability at this data center
- Integer value for every feature that can be built representing the amount of work put into building that feature.
- Integer value for every feature that can be built representing the amount of work required for building that feature.
- Work Target - int representing which feature is being worked on.
- Resources: int - The amount of resources being used for the production of this data center
- Date of scan: date - the time of the last network scan.

6.4 Malware

The following information is the data regarding malware that is to be stored when the user saves the game.

- ID: int - the ID of the malware.
- Name: String - the name of the malware.
- Malware Type: String - the type of malware used
- List of int values for each attribute of the malware (stealth, speed, size, etc.)
- List of boolean values for each special attribute the malware can have.
- Resources Rate: int - rate of production
- Resources Produced: int - number of resources put into the production of this.
- Required Resources: int - number representing how much work is needed to produce this

6.5 Attacks

The following information is the data regarding attacks that is to be stored when the user saves the game.

- ID: int - the ID of the attack.

- Name: String - the name of the attack.
- Malware: int - the ID of the malware used for the attack (0 will always be reconnaissance).
- Owner: int - player ID of the owner of the malware.
- Target: int - data center ID of the target data center.
- Mode of Delivery: String representing how the malware is being delivered.
- Exploit: int - number representing an existing vulnerability in the target data center that the attack will be exploiting.
- Goal: String representing what the goal of the attack is (steal money, create backdoor, etc.)
- Production Rate: rate of production.
- Resources Produced: int - number of resources put into the production of this.
- Required Resources: int - number representing how much work is needed to produce this.

7. Timeline

Week Number	**Milestone**
Week 1 - May 20th	Project Initialization
Week 2 - May 27th	Core Gameplay Mechanics
Week 3 - June 3rd	Offensive Mechanics
Week 4 - June 10th	
Week 5 - June 17th	Defensive Mechanics
Week 6 - June 24th	Special Features
Week 7 - July 1st	Menus
Week 8 - July 8th	Data Storage
Week 9 - July 15th	
Week 10 - July 22nd	Art & Design
Week 11 - July 29th	Computer Players
Week 12 - August 5th	Audio
Week 13 - August 12th	Polish
Week 14 - August 19th	P2P