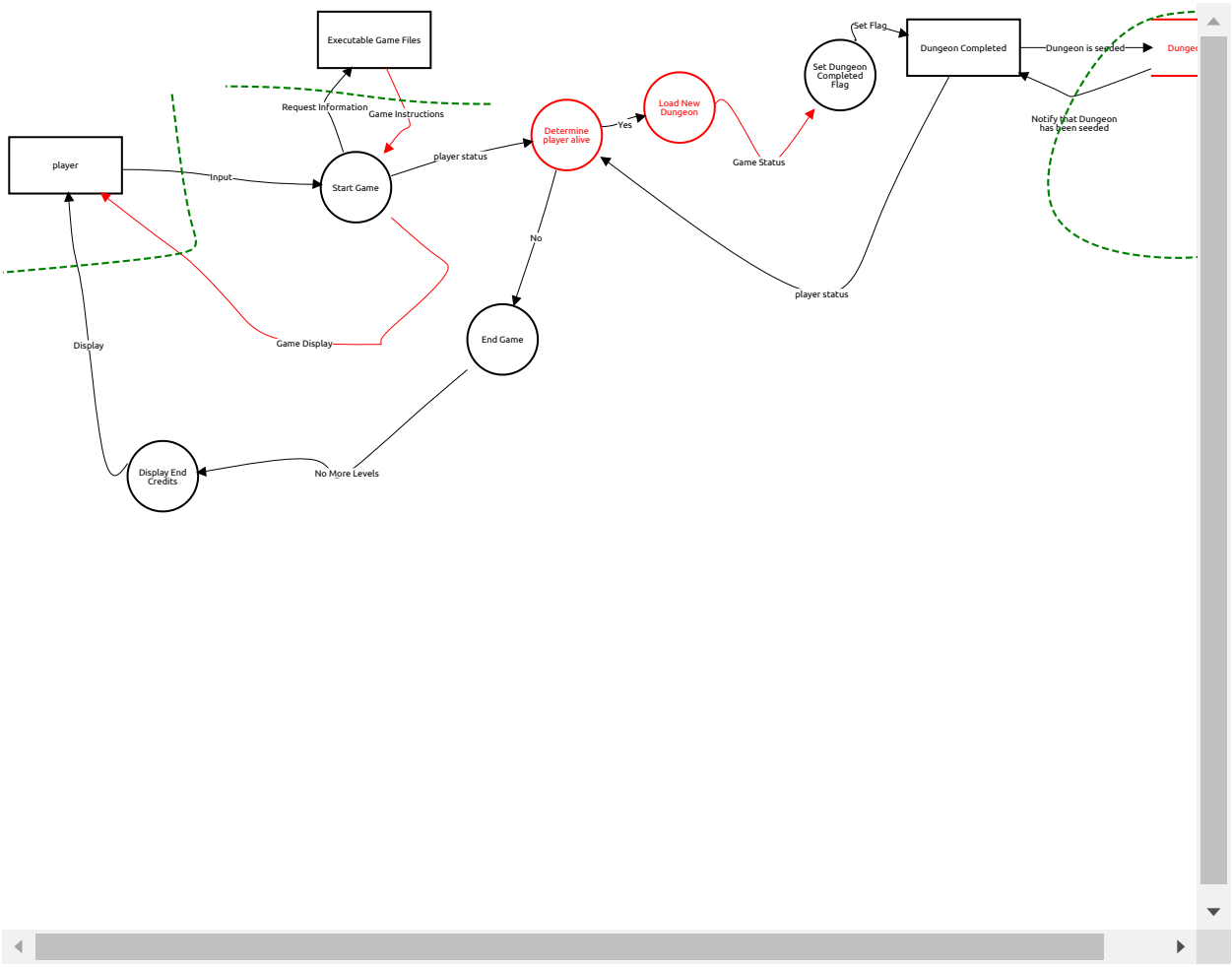
Threat model report for Software Engineering Assignment 2

Owner:
Gbadebo Adeyela
Reviewer:
Michael Schukat
Contributors:

High level system description

DFD Model for Final Year Project that represents a roguelike dungeon crawler game that implements procedural content generation

PCG Game DFD


Executable Game Files
Request Information
Game Instructions
player
Input
Start Game
player status
Determine player alive
Yes
Load New Dungeon
Game Status
Set Dungeon Completed Flag
Set Flag
Dungeon Completed
Dungeon is seeded
Notify that Dungeon has been seeded
player status
No
End Game
Display
Game Display
Display End Credits
No More Levels

Start Game (Process)

Description:
Game is ran

Malware tampering threat

Tampering, Mitigated, Medium Severity

Description:
Game can be tampered with Malware at runtime . This is escpecially true for third party publishers of the game.

Mitigation:
The introduction of authorization and authentication mechanisms for players should prevent tampering by malware

Trojan information disclosure threat

Information disclosure, Mitigated, Medium Severity

Description:
Trojans with keyboard recording function can be deployed at runtime in order to record the keystrokes and the sequence of the user’s actions

Mitigation:
Cryptographic protocols are deployed to battle hackers who implements Trojan hacks

Local File Inclusion information disclosure threat

Information disclosure, Mitigated, High Severity

Description:
Local File Inclusion can deployed at runtime of the game by hackers to gain access to files stored on the server

Mitigation:
Add firewall to server to prevent Local File Inclusion from being deployed

player (External Actor)

Description:
User whos action determine game generation

Bot spoofing threat

Spoofing, Mitigated, Medium Severity

Description:
User could use bot in order to do well in the game.

Mitigation:
Add a Captcha test. As a Captcha test effeciently removes bots that cannot read and supply a correct answer to the test

No Login repudiation threat

Repudiation, Mitigated, Medium Severity

Description:
The lack of a login system in the game means that the game does not adopt controls to properly track and log users' actions, thus permitting malicious manipulation

Mitigation:
Add a login system for the game to track and log users' actions in order to prevent malicious manipulation

Determine player alive (Process)

Description:
Game determines if player is still alive

Player Health tampering threat

Tampering, Open, Medium Severity

Description:
Hackers could attempt to modify player character's health so that they never die and the game never ends

Mitigation:

player status (Data Flow)

Description:
The status of the player's health

No threats listed.

Executable Game Files (External Actor)

Description:
Source code of game

Admin spoofing threat

Spoofing, Mitigated, Medium Severity

Description:
User/bot who illegally gain access to admin rights can falsely pretend to be an admin and gain access to game's source code

Mitigation:
Require two-factor authentication tp gain access to admin right so that it is more difficult for hackers to gain access to source code

Request Information (Data Flow)

Description:
Game requested for info from executable files

No threats listed.

Game Instructions (Data Flow)

Description:
Instructions are sent so game can be run

Game Instructions tampering threat

Tampering, Open, Medium Severity

Description:
Game Instructions sent by game could be tampered with by hacker in order to meet hacker's desire

Mitigation:

Instructions information disclosure threat

Information disclosure, Open, Medium Severity

Description:
Inforamtion about the Instructions could possiby be accquired by hackers to manipulate the game for nefarious purposes

Mitigation:

Load New Dungeon (Process)

Description:
A new dungeon is loaded if player is still alive

Manipulating Dungeon Generation elevation threat

Elevation of privilege, Open, Medium Severity

Description:
Hackers could potentially manipulate the game's dungeon generation to continuously generate dungeons in order to eventually crash the player's system

Mitigation:

Yes (Data Flow)

Description:
Player is alive

No threats listed.

Set Dungeon Completed Flag (Process)

Description:
Flag is set for dungeon completed

No threats listed.

Game Status (Data Flow)

Description:
Determine the current status of game

Game Status DoS threat
Denial of service, Open, Low Severity

Description:
The game status could be prevented from updatibg thus preventing dungeoun generation

Mitigation:

Game Status tampering threat
Tampering, Open, Medium Severity

Description:
Status of Game could be tampered with in order to perfom a DoS threat

Mitigation:

Dungeon Completed (External Actor)

Description:
Player has completed the dungeon

No threats listed.

Set Flag (Data Flow)

Description:
Flag is set

No threats listed.

Display End Credits (Process)

Description:

No threats listed.

End Game (Process)

Description:
The end of the game has been reached as player is dead

No threats listed.

No (Data Flow)

Description:
Player is dead

No threats listed.

No More Levels (Data Flow)

Description:
Level Generation is halted after game has ended

No threats listed.

Display (Data Flow)

Description:
End Credits are displayed to player

No threats listed.

Game Display (Data Flow)

Description:
Game is diplayed to the player

Display tampering threat

Tampering, Open, Medium Severity

Description:
Hackers could attempt to manipulate the game's display for their own nefarious puposes

Mitigation:

Input (Data Flow)

Description:
Action inputted by Player

No threats listed.

Dungeon Database (Data Store)

Description:
Database for completed dungeons

SQL injection tampering threat
Tampering, Mitigated, High Severity

Description:
Hackers tamper with the database by using online forms to inject specific SQL code that can then compromise the database

Mitigation:
Input validation and parametrized queries including prepared statements should be used to prevent SQL Injection attack

Dungeon Generation DoS threat
Denial of service, Open, High Severity

Description:
Hackers can deny service of game by not allowing the database to notify the game that dungeon has been seeded thus halting dungeon generation

Mitigation:

Seeded Dungeon information disclosure threat
Information disclosure, Open, High Severity

Description:
Gaining access to database of seeded dungeons could provide hackers with vital info on how dungeons are generated thus giving further insight on how to manipulate game

Mitigation:

Dungeon is seeded (Data Flow)

Description:
Dungeon is seeded so it is not repeated for generation

No threats listed.

Notify that Dungeon has been seeded (Data Flow)

Description:
Notify the game that completed dungeon has been added to database

No threats listed.

player status (Data Flow)

Description:
The status of the player's health

No threats listed.