

Cyber Security Education

User Manual

By Liana Villafuerte, Bruce Wagner, Kyle Stead, and
Hussein Okasha

Table of Contents:

Table of Contents	2 - 3
Cyber Security Education Goal	4
Cyber Security Education Challenges	4
Disclaimer and Warnings	5
Game Story	5 - 6
Game Controls	6 - 7
Game Menu	8 - 10
Section 1: Name Input	8
Section 2: Level Input	9
Section 3: Controls Button	9
Section 4: Start Button	9
Section 5: End Button	10
Pause Menu	10 - 12
Section 1: Resume	11
Section 2: Controls	11
Section 3: Restart	11
Section 4: Main Menu	11

Game Entities	12 - 13
Section 1: Main Character	12
Section 2: Turrets	13
Section 3: Cores	13
Zipped Folders	14 - 15
Section 1: Dist Folder	14
Section 2: Answer Folder	15
How to Contact Us	16

Cyber Security Education Goal

_____The goal is to create a form of practice and introduction of cyber security to college students who wish to join the Florida Institute of Technology Cyber Security team. This will be done through a hackable shooter game that has various vulnerabilities that students will exploit for their advantage. An example in our program is having the user observe a cipher and decipher it so they can implement a vulnerability that allows them to be invisible. The motivation for this is so that college students can practice their cyber security skills when they join in hacking tournaments throughout the year and see if this would be a career field they would like to pursue

Students will learn the concepts and tools of cyber security by participating in FITSEC and learning how to exploit the game. The mechanics that will be used to achieve this consist of having the student reverse engineer the game and find vulnerabilities they can use to win it. Challenges will consist of finding ways to exploit the game and complete levels and objectives in order to either beat the game. Game mechanics will include sql injection, cryptography, directory traversal, break access control, and reverse engineering.

Cyber Security Education Challenge

_____Not all the members in the group have used python at advanced levels. Some concepts also require more time to implement than others, due to the differences in difficulty and code needed to implement it. Keeping the integrity of the code architecture while successfully implementing all the necessary concepts will also be difficult. The hardest part was figuring out how to add extra details to the game that are simple and basic to understand.

Some of these challenges were solved by using Pygames, since it allows us to not just have the game as a normal application but still remain vulnerable to outside manipulation. We also give the users a clear and interactive understanding of the game and how they can go about exploiting various aspects through hints. We also organized and made it easier to search functions while students look through the code for clues. This helped us create a game that is an

easy introduction for new FITSEC recruits to practice their skills and get used to finding exploits while learning with the team.

Disclaimer and Warnings

_____This game was not originally made by us. We went out and found a decently made shooter game made by a developer going by the name of dafluffypotato. We have included their contact details at the end, and included their license and notice. We are responsible for building and changing the code and game so as to fit the necessary elements required to make this an engaging cyber security game.

Game Story

_____In the year 2030, large tech companies were performing inhumane tests on people, while the customers up above used the safe versions to enhance their life. The company tested with different variations of implants, ranging from replacing an arm with a robotic one that can extend and retract at will, adding spring coil system in both legs to allow super jumping abilities, and adding an implant to the brain that acts as an internal processor to think at super speeds and allows them the ability to teleport short distances when in danger. All the test subjects along the years had either failed or were thrown out, except one. Glitch, an 18 year old, was tested and monitored for years on end, having been one of the most successful experiments to date. One day, Glitch escaped and swore revenge on the company that scarred them for life. The company took no heed of his vengeance, because they were so confident in their security turrets. The tech company was so confident in themselves, they called their company "Invulnerable Labs".

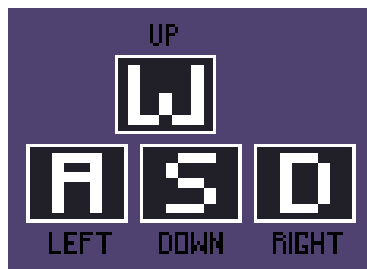
Fast forward to the year 2040, the implants started improving and being sold in large batches, slowly being seen as a symbol of superiority; something Glitch despised with such venom. Those very tech items were the reason for his suffering. With each passing day, he hid further into his cloak, scowling to those of the upper class. Having spent all his time hunting down the location

of the laboratory, Glitch enacted his revenge. Glitch, given what they learned in 20 years living with their implants, finally has the chance to put a stop to them. They picked up their shotgun and decided to destroy the lab from the inside out and exploit their weaknesses, no matter the cost. For what they didn't realize, was they had Vulnerable Lairs.

Game Controls

WASD Keys: The WASD keys are the input keys responsible for movement around the map. The “W” key, along with the spacebar, are responsible for the jumping motion in the character. Which allows the player to jump onto a ledge or just move around the map. The “A” and “D” keys are responsible for respectively moving the character left and right of the map. The “S” key is registered to go down, but there is no crouch mechanic. The visual diagram can be seen in figure 1.

Figure 1.



Left mouse button: The left mouse button allows the player to shoot their shotgun either at the cores or the turrets for destruction. Players can even shoot at the ground and this will allow them some recoil and it will lift them into the air. The visual diagram for the control can be seen in figure 2.

Figure 2.



Right mouse button: The right mouse is in control of the player's ability to slow down time and the map and to zoom in to assess the targets and eliminate them. This ability is only used when the player holds the right button. The visual diagram can be seen in figure 3.

Figure 3.



Space Bar: The space bar, along with the “W” key, are responsible for the player's ability to jump across the map either over ledges or just general navigation across the map. The visual diagram for the control can be seen in figure 4.

Figure 4.



Mouse Tracker: The mouse movements are used as an aiming system for the player's shotgun to aim at the different targets. The visual diagram for the control can be seen in figure 5.

Figure 5.



Game Menu

_____The game menu helps the user start up the game and direct them into either simply starting the game, or helps them view the controls of the game. The game menu contains a name input text field, a level input text field, start button, controls button, and an end button as shown in figure 6.

Figure 6.



Section 1: Name Input

_____This is the text field where the user puts their name which is later on used to store their overall score in the scores database, just like in an arcade game where it stores the name and the highest score of that name. This field is also used for one or more exploits in the game.

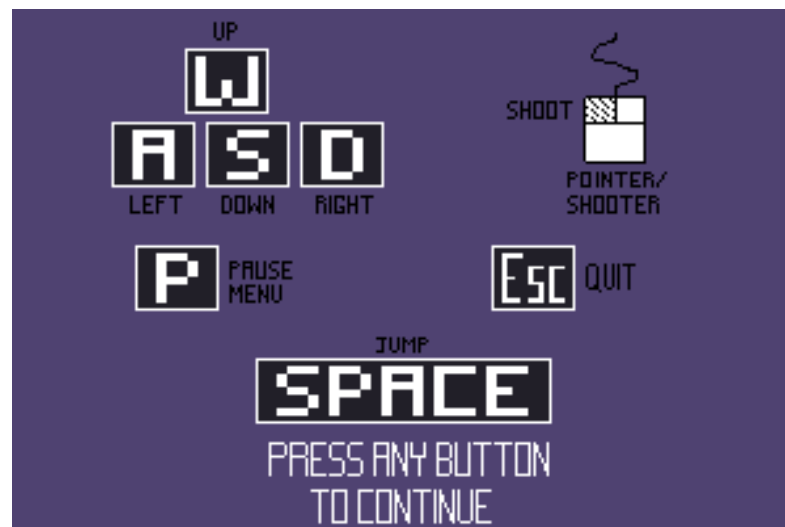
Section 2: Level Input

_____ This is the text field that the user inputs the level they want to work on or get to. If left empty or is given a level that isn't within the range of available levels, the game will automatically start at level 1 as the default. This field is also used for one or more exploits in the game.

Section 3: Controls Button

_____ Clicking this button displays the controls menu which informs the User how to move and play the game, it can be accessed either from the main menu or from the pause menu, as shown in figure 7 as an overall view.

Figure 7.



Section 4: Start Button

_____ This button starts the game with the instructions given from the user such as the level desired and the name they want to store in the database. If both are empty, it stores the level under an empty name and proceeds as normal from level 1.

Section 5: End Button

_____This button exits the game, along with the “Esc” key, it can be accessed either from the pause menu or from the main menu. This button was created so as to allow the user to manually escape if their keyboard’s esc button doesn’t work.

Pause Menu

Pause menu is accessed from mid level game play by clicking the “P” key at any point mid game. This will freeze the game and it will continue the game normally when the player presses “P” again, as shown in Figure 8.

Figure 8.



Section 1: Resume

When the game pauses, the screen freezes all the entities in the game. The resume button continues the game from the paused point and allows the user to resume the game from wherever they paused mid game.

Section 2: Controls

As previously mentioned in Game Menu: Section 3, this button displays the controls of the game in case the user forgets the controls to move around the map, jump, or shoot their weapon.

Section 3: Restart

Clicking this button restarts the level from the very beginning and resets the turrets, core count, and the player's position back to spawn point as if they just started the level. You will hear a dying sound since this was the most efficient way of restarting.

Section 4: Main Menu

This button sends the user to the main menu which can allow them to change the level they are on. For more information on the main menu, please refer to the Game Menu Section.

Game Entities

The characters in this game all have different uses in the game regarding playability, goals, and user interaction. The three entities that help make up this game are the turrets, cores, and main character, Glitch.

Section 1: Main Character

The main character, Glitch, is a cloaked figure the user plays as. Glitch is equipped with multiple abilities due to what Invulnerable Labs did to him. He has the ability to jump high, cybernetic eyes to help aim his shotgun at cores, slow time when near the turret's bullets, and the ability to turn invisible, if the user helps him. The player must help Glitch through the different lairs in order to destroy the cores that power the lab. Or, the user could learn how to manipulate the game by exploiting its vulnerabilities and helping Glitch find the secret core room. Figure 9 below shows what Glitch the Character looks like.

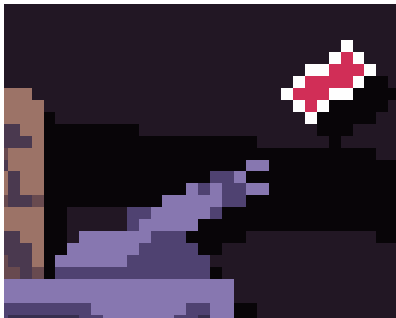
Figure 9.



Section 2: Turrets

The turrets are the security system of the invulnerable labs. The turrets are meant to keep people like the main player away from the cores. Turrets aim at the player and have a cooldown of 1 second before firing their next shot at the player's new location. Once it shoots, the bullet proceeds in a straight linear path. They can, however, be shot at and destroyed with a bullet, as long as you don't get shot first. Figure 10 below shows a turret mid shooting the player.

Figure 10.



Section 3: Cores

The cores are what power the lab, and are scattered throughout the lairs. One shot will be able to destroy them. Some cores are placed near each other while others are hard to reach due to the amount of turrets surrounding them and the terrain. In order to progress, the player must beat all of the cores in the level.

Figure 11.



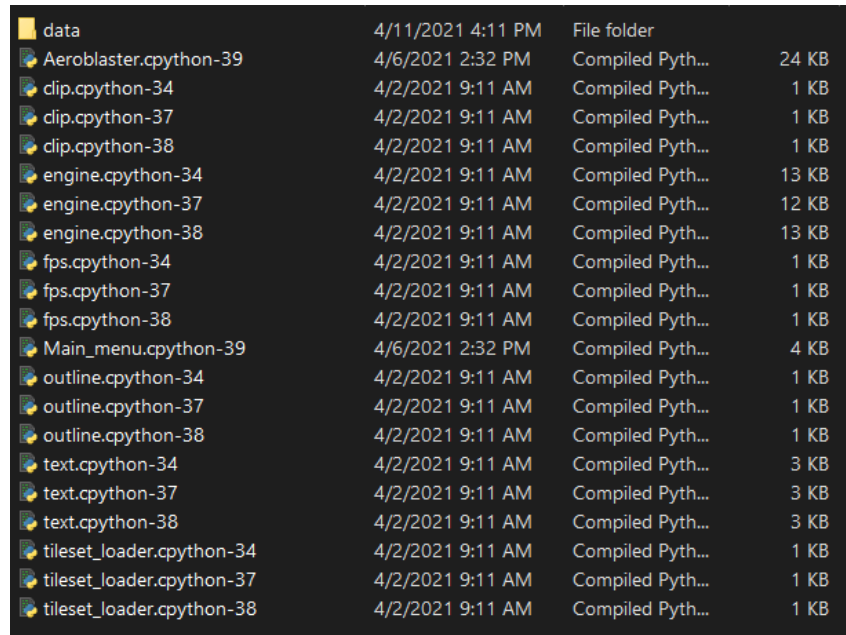
Zipped Folders

This section describes what to expect in each of the zipped folders included in this project. Both have their distinct function, and are useful to both students and teachers alike. Also as a disclaimer, while the dist folder is unlocked, the answer folder is locked and zipped by 7-zip, and can be unlocked that way with the proper password. This is to ensure students don't gain access unless they perform break access methods, and to also make it so teachers who distribute the game, can also just remove it and keep it for their own personal use.

Section 1: Dist Folder

The dist folder contains .pyc files of the game and its needed data folder. .pyc files are bytecode files of python programs. This is so that if students would like to challenge themselves to reverse engineer bytecode instead of just normal pythonic code. This works due to the fact that .pyc files are cythonized python files that can be analyzed just like normal binary code. It can also be run by python using the command “python3 Main_menu.cpython-39”. Figure 12 on the next page shows the contents of the folder after being unzipped. If corrupted, the files can be replicated by running “python3 -m compileall .” and copying the data folder into the new compiled folder called “__pycache__”.

Figure 12.



data	4/11/2021 4:11 PM	File folder	
Aeroblaster.cpython-39	4/6/2021 2:32 PM	Compiled Pyth...	24 KB
clip.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
clip.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
clip.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
engine.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	13 KB
engine.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	12 KB
engine.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	13 KB
fps.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
fps.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
fps.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
Main_menu.cpython-39	4/6/2021 2:32 PM	Compiled Pyth...	4 KB
outline.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
outline.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
outline.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
text.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	3 KB
text.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	3 KB
text.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	3 KB
tileset_loader.cpython-34	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
tileset_loader.cpython-37	4/2/2021 9:11 AM	Compiled Pyth...	1 KB
tileset_loader.cpython-38	4/2/2021 9:11 AM	Compiled Pyth...	1 KB

Section 2: Answer Folder

The answer folder contains three images. All the images contain the needed input to exploit the name and level input with the SQL injection, cryptography, and directory traversal vulnerabilities. In order to access this document, teachers must enter the password “246” While opening it with 7-Zip. Students, on the other hand, must use the hint and resources provided, to try and break the access to the folder if they wish to try and take the “easier” route.

How to Contact Us

If there are any further questions about the concepts of the game, the best contact is the Project lead and group. For questions regarding the game or permissions to its basic, the best contact is the Licensed Owner.

Project Lead and group:

Liana Villafuerte: lvillafuerte2018@my.fit.edu

Bruce Wagner: wagnerb2016@my.fit.edu

Kyle Stead: kstead2016@my.fit.edu

Hussein Okasha: hokasha2016@my.fit.edu

Licensed Owner of Original Game:

Website: <https://dafluffypotato.com/>

Email: dafluffypotatodev@gmail.com

Twitter: <https://twitter.com/DaFluffyPotato>