# CS 410 Project Two Security Report Template

## Instructions

Fill in the table in step one. In steps two and three, replace the bracketed text with your answer in your own words.

Identify where multiple security vulnerabilities are present within the blocks of C++ code. You may add columns and extend this table as you see fit.

| **Block of C++ Code** | **Identified Security Vulnerability** |
| --- | --- |
| void ChangeCustomersChoice(bool access) {  int changeChoice;  int changeChoice2;  cout << "Enter the number of the client that you wish to change\n";  cin >> changeChoice; | There is no input validation system. If a incorrect choice is done, the program throws itself into a loop. |
| while (access == true) {  cout << "What would you like to do?\n";  cout << "DISPLAY the client list (enter 1)\n";  cout << "CHANGE a client's choice (enter 2)\n";  cout << "Exit the program .. (enter 3)\n";  cin >> choice; | This is another spot where there is no choice validation. When a letter is input, the program breaks and goes into a loop of calling the DisplayInfo() function. |
| bool CheckUserPermissionAccess(string userName, string password) {  bool grantAccess = false;  while(1) {  if (userName == "izzy" && password == "123") {  grantAccess = true;  return grantAccess;  }  else {  return grantAccess;  }    } | This is not a secure way of giving access to a program. There is no encryption to the password. Any hacker could just reverse engineer the code and see the password and username in plain text. |
| string name1 = "Bob Jones";  string name2 = "Sarah Davis";  string name3 = "Amy Friendly";  string name4 = "Johnny Smith";  string name5 = "Carol Spears";  int num1 = 1;  int num2 = 2;  int num3 = 1;  int num4 = 1;  int num5 = 2; | These variables are global variables. The use of global variables can cause bugs and are a pollution of the namespace. |

2.Explain the *security vulnerabilities* that are found in the blocks of C++ code.

[In a paragraph or two for each security vulnerability, explain in detail how and why these are security vulnerabilities.]

When looking at the first block of code, when the code is taken in the user’s choice, there needs to be a validation system for what the user has input. This causes the code to break when a letter is input. This break causes the DisplayInfo function to be called infinitely not letting the user access the rest of the program.

The second block of code above also has an issue with the input validator being missing. If the user puts in a number higher than the choices it will just do nothing and recall the DisplayInfo function. But like above, when a letter is input, the program breaks into the same recuring loop.

When the program is validating the password and username, the information is hard coded into the program. This is a serious violation of the security of the program. Any person who knows how to read code can easily hack the C++ file and see in the code what the password and username are, and log into the program with ease.

The variables that are storing the information of the clients are all stored in global variables. This could cause bugs and later issues in the program if a variable is used as a local variable is named the same as the global variable. It’s important to not use global variables as much in programs, try to use local ones.

Describe *recommendations* for how security vulnerabilities can be fixed.

[In a paragraph or two for each recommendation, describe how you would fix these vulnerabilities.]

To fix the code for the first block of code, there should be an input validation system to make sure what is being input and checks it before it is just taken in. The main issue is that the input that is being taken in if it’s a letter the code breaks and starts to go into a recuring loop of the Display Info function. A way to fix this issue is when taking in the input, make sure after that if it’s a letter or string, to clear the memory and tell the user to pick something else. This needs to be done to both blocks of code that take in the users input for navigation purposes. This would be done with the password one since the username and password will be strings.

When looking at the password checker, it suffers from having the username and password hard coded into the program. This would be an issue due to a hacker even a beginner can see the blatant information right In the C code. There should be encryption to prevent the password from leaking out. Another way of fixing the issue could be storing the password outside the program so it’s not so easy to locate. Using a simple import of a text document could help by storing the information off the program. There could also be a encryption to the data with a key so it makes it almost impossible to crack the password.

When coding a program, it’s bad practice on using global variables. They can case bugs and issues later in coding by polluting the namespace. When a program gets longer and more complex there could be times where the same name of a global variable is used, and that would cause major issues since now the variables are changing both variables. This can be rectified by using pointers. By pointing to the specific location instead of the name, there won’t be any instances when the variables will be used twice.