# **CS 410 Binary to C++ With Security Vulnerabilities Activity Template**

**Step 1:** Convert the binary file to assembly code.

**Step 2:** Explain the functionality of the blocks of assembly code.

| **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- |
| <+33>: cmp $0x5,%eax  <+36>: je 0xd02 <main+655> | While the number entered by the user is not 5 the program iterates, once 5 is entered then the program jumps to the exit functions |
| <+49>: lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3a4(%rip),%rsi # 0xe5b  <+68>: lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x39c(%rip),%rsi # 0xe66  <+87>: lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x399(%rip),%rsi # 0xe76  <+106>: lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: lea 0x396(%rip),%rsi # 0xe86  <+125>: lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+132>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+137>: lea 0x346(%rip),%rsi # 0xe49  <+144>: lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+151>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Loads the strings and prints the menu of choices for the user to see   1. Add 2. Subtract 3. Multiply 4. Exit |
| <+163>: lea 0x201623(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+170>: callq 0x870 <\_ZNSirsERi@plt> | Accepts a user input for the menu choice |
| <+175>: mov -0x14(%rbp),%eax  <+178>: cmp $0x1,%eax  <+181>: jne 0xbc3 <main+336> | Checks if input is 1 if its not moves to the next value check against input |
| <+194>: lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+201>: callq 0x870 <\_ZNSirsERi@plt>  <+219>: callq 0x870 <\_ZNSirsERi@plt>  <+224>: mov -0x10(%rbp),%eax  <+290>: mov -0x10(%rbp),%edx  <+293>: mov -0xc(%rbp),%eax  <+296>: sub %eax,%edx  <+305>: callq 0x8d0 <\_ZNSolsEi@plt>  <+331>: jmpq 0xa91 <main+30> | Accepts user input for first number, then a second number. Performs a subtraction calculation on it and out puts “first number – second number = answer” and returns to the menu |
| <+339>: cmp $0x2,%eax  <+342>: jne 0xc62 <main+495> | Checks if the user input for menu selection is 2, if not jump s to next menu input comparison |
| <+355>: lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+362>: callq 0x870 <\_ZNSirsERi@plt>  <+380>: callq 0x870 <\_ZNSirsERi@plt>  <+457>: add %edx,%eax  <+459>: mov %eax,%esi  <+461>: mov %rcx,%rdi  <+464>: callq 0x8d0 <\_ZNSolsEi@plt>  <+490>: jmpq 0xa91 <main+30> | Accepts input for a first number then a second number. Adds the two numbers together and prints the out put of “firstNumber – secondNumber = answer”  Returns user to the menu. |
| <+498>: cmp $0x3,%eax  <+501>: jne 0xa91 <main+30> | Checks if user input for menu selection is 3, if not loads the menu again |
| <+514>: lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+521>: callq 0x870 <\_ZNSirsERi@plt>  <+539>: callq 0x870 <\_ZNSirsERi@plt>  <+616>: cltd  <+617>: idiv %esi  <+619>: mov %eax,%esi  <+621>: mov %rcx,%rdi  <+624>: callq 0x8d0 <\_ZNSolsEi@plt> | Accepts input for first number. Accepts input for second number. Divides the two numbers. Prints “firstNumber – secondNumber = answer” |

**Step 3:** Convert the assembly code to binary.

**Step 4:** Convert the assembly code to C++ code.

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| <+33>: cmp $0x5,%eax  <+36>: je 0xd02 <main+655> | While (userChoice != 5) { |
| <+49>: lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3a4(%rip),%rsi # 0xe5b  <+68>: lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x39c(%rip),%rsi # 0xe66  <+87>: lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x399(%rip),%rsi # 0xe76  <+106>: lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: lea 0x396(%rip),%rsi # 0xe86  <+125>: lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+132>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+137>: lea 0x346(%rip),%rsi # 0xe49  <+144>: lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+151>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Cout << “---------------” << endl;  Cout << “ – 1) Add –“ << endl;  Cout << “ – 2) Subtract –“ << endl;  Cout << “ – 3) Multiply –“ << endl;  Cout << “ – 4) Exit –“ << endl;  Cout << “---------------” << endl; |
| <+175>: mov -0x14(%rbp),%eax  <+178>: cmp $0x1,%eax  <+181>: jne 0xbc3 <main+336>  <+194>: lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+201>: callq 0x870 <\_ZNSirsERi@plt>  <+219>: callq 0x870 <\_ZNSirsERi@plt>  <+224>: mov -0x10(%rbp),%eax  <+290>: mov -0x10(%rbp),%edx  <+293>: mov -0xc(%rbp),%eax  <+296>: sub %eax,%edx  <+305>: callq 0x8d0 <\_ZNSolsEi@plt>  <+331>: jmpq 0xa91 <main+30> | If (menuChoice == 1) {  Cin >> firstNumber;  Cout << endl;  Cin >> secondNumber;  Cout << endl;  Answer = firstNumber – seconmdNumber;  Cout << firstNumber << “ – “ << secondnumber  << “ = “ << answer << endl;  } |
| <+339>: cmp $0x2,%eax  <+342>: jne 0xc62 <main+495>  <+355>: lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+362>: callq 0x870 <\_ZNSirsERi@plt>  <+380>: callq 0x870 <\_ZNSirsERi@plt>  <+457>: add %edx,%eax  <+459>: mov %eax,%esi  <+461>: mov %rcx,%rdi  <+464>: callq 0x8d0 <\_ZNSolsEi@plt>  <+490>: jmpq 0xa91 <main+30> | Else If (menuChoice == 2) {  Cin >> firstNumber;  Cout << endl;  Cin >> secondNumber;  Cout << endl;  Answer = firstNumber + secondNumber;  Cout << firstNumber << “ – “ << secondnumber  << “ = “ << answer << endl;  } |
| <+498>: cmp $0x3,%eax  <+501>: jne 0xa91 <main+30>  <+514>: lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+521>: callq 0x870 <\_ZNSirsERi@plt>  <+539>: callq 0x870 <\_ZNSirsERi@plt>  <+616>: cltd  <+617>: idiv %esi  <+619>: mov %eax,%esi  <+621>: mov %rcx,%rdi  <+624>: callq 0x8d0 <\_ZNSolsEi@plt> | Else If (menuChoice == 3) {  Cin >> firstNumber;  Cout << endl;  Cin >> secondNumber;  Cout << endl;  Answer = firstNumber / secondNumber;  Cout << firstNumber << “ – “ << secondnumber  << “ = “ << answer << endl;  } |