# **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** |
| --- | --- |
| 0x0000000000000079 <+0>: push %rbp  0x000000000000007a <+1>: mov %rsp,%rbp  0x000000000000007d <+4>: sub $0x20,%rsp  0x0000000000000081 <+8>: mov %fs:0x28,%rax  0x000000000000008a <+17>: mov %rax,-0x8(%rbp) | Saves initial stack then moves %rsp to %rbp then subtracts 0x20 by %rsp  After that it moves fs:0x28 to %rax and then moving %rax to %rbp |
| 0x00000000000000a3 <+42>: lea 0x0(%rip),%rsi # 0xaa <main+49>  0x00000000000000aa <+49>: lea 0x0(%rip),%rdi # 0xb1 <main+56>  0x00000000000000b1 <+56>: callq 0xb6 <main+61>  0x00000000000000b6 <+61>: lea 0x0(%rip),%rsi # 0xbd <main+68>  0x00000000000000bd <+68>: lea 0x0(%rip),%rdi # 0xc4 <main+75>  0x00000000000000c4 <+75>: callq 0xc9 <main+80>  0x00000000000000c9 <+80>: lea 0x0(%rip),%rsi # 0xd0 <main+87> | Starts by loading the address in memory twice and then executes with the callq. It does this about 3 times in this block |
| 0x0000000000000128 <+175>: mov -0x14(%rbp),%eax  0x000000000000012b <+178>: cmp $0x1,%eax  0x000000000000012e <+181>: jne 0x1c9 <main+336>  0x0000000000000134 <+187>: lea -0x10(%rbp),%rax  0x0000000000000138 <+191>: mov %rax,%rsi | This block starts by moving %rbp to eax and then compares 0x1 and eax. After that we jump to location 0x1c9 and then load the address there with the lea and finally move %rax to %rsi |
|  |  |
|  |  |

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

Linked with zip

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

CPP file with zip

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| a73: 55 push %rbp  a74: 48 89 e5 mov %rsp,%rbp  a77: 48 83 ec 20 sub $0x20,%rsp  a7b: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax  a82: 00 00  a84: 48 89 45 f8 mov %rax,-0x8(%rbp)  a88: 31 c0 xor %eax,%eax  a8a: c7 45 ec 00 00 00 00 movl $0x0,-0x14(%rbp)  a91: 8b 45 ec mov -0x14(%rbp),%eax  a94: 83 f8 05 cmp $0x5,%eax  a97: 0f 84 65 02 00 00 je d02 <main+0x28f> | if(input == 1){  int input1;  int input2; |
| a9d: 48 8d 35 a5 03 00 00 lea 0x3a5(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  aa4: 48 8d 3d 75 15 20 00 lea 0x201575(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  aab: e8 e0 fd ff ff callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  ab0: 48 8d 35 a4 03 00 00 lea 0x3a4(%rip),%rsi # e5b <\_ZStL19piecewise\_construct+0x57>  ab7: 48 8d 3d 62 15 20 00 lea 0x201562(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  abe: e8 cd fd ff ff callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  ac3: 48 8d 35 9c 03 00 00 lea 0x39c(%rip),%rsi # e66 <\_ZStL19piecewise\_construct+0x62>  aca: 48 8d 3d 4f 15 20 00 lea 0x20154f(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  ad1: e8 ba fd ff ff callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  ad6: 48 8d 35 99 03 00 00 lea 0x399(%rip),%rsi # e76 <\_ZStL19piecewise\_construct+0x72>  add: 48 8d 3d 3c 15 20 00 lea 0x20153c(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  ae4: e8 a7 fd ff ff callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  ae9: 48 8d 35 96 03 00 00 lea 0x396(%rip),%rsi # e86 <\_ZStL19piecewise\_construct+0x82>  af0: 48 8d 3d 29 15 20 00 lea 0x201529(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4> | cout << input1 << " + " << input2 << " = " << input1 + input2 << endl;  }else if(input == 2){  int input3;  int input4;  cout << input3 << " - " << input4 << " = " << input3 - input4 << endl; |
| b0f: 48 8d 45 ec lea -0x14(%rbp),%rax  b13: 48 89 c6 mov %rax,%rsi  b16: 48 8d 3d 23 16 20 00 lea 0x201623(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  b1d: e8 4e fd ff ff callq 870 <\_ZNSirsERi@plt>  b22: 8b 45 ec mov -0x14(%rbp),%eax  b25: 83 f8 01 cmp $0x1,%eax  b28: 0f 85 95 00 00 00 jne bc3 <main+0x150>  b2e: 48 8d 45 f0 lea -0x10(%rbp),%rax  b32: 48 89 c6 mov %rax,%rsi | else if(input == 3){  int input5;  int input6; |
| b35: 48 8d 3d 04 16 20 00 lea 0x201604(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  b3c: e8 2f fd ff ff callq 870 <\_ZNSirsERi@plt>  b41: 48 89 c2 mov %rax,%rdx  b44: 48 8d 45 f4 lea -0xc(%rbp),%rax  b48: 48 89 c6 mov %rax,%rsi  b4b: 48 89 d7 mov %rdx,%rdi  b4e: e8 1d fd ff ff callq 870 <\_ZNSirsERi@plt>  b53: 8b 45 f0 mov -0x10(%rbp),%eax  b56: 89 c6 mov %eax,%esi | cout << input5 << " \* " << input6 << " = " << input5 \* input6 << endl;  }else if(input == 4){  exit(0); |
|  |  |