# CS-410 Software Reverse Engineering – Assignment 6

Binary to C++ With Security Vulnerabilities

Eric Slutz

Southern New Hampshire University

| Blocks of Assembly Code | Explanation of Functionality |
|---|---|
| push %rbp | Start program and begin looping until %eax comparison to 5 is true. |
| mov %rsp,%rbp | |
| sub $0x20,%rsp | |
| mov %fs:0x28,%rax | |
| mov %rax,-0x8(%rbp) | |
| xor %eax,%eax | |
| movl $0x0,-0x14(%rbp) | |
| mov -0x14(%rbp),%eax | |
| cmp $0x5,%eax | |
| je 0x308 <main+655> | |
| lea 0x0(%rip),%rsi | Displays menu |
| lea 0x0(%rip),%rdi | |
| callq 0xb6 <main+61> | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0xc9 <main+80> | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0xdc <main+99> | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0xef <main+118> | |
| lea 0x0(%rip),%rsi | Gets menu selection |
| lea 0x0(%rip),%rdi | |
| callq 0x102 <main+137> | |
| lea 0x0(%rip),%rsi | Gets first user input |
| lea 0x0(%rip),%rdi | |
| callq 0x115 <main+156> | |
| lea -0x14(%rbp),%rax | Gets second user input |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x128 <main+175> | |
| mov -0x14(%rbp),%eax | Performs subtraction on user input and outputs the result in a formatted statement. |
| cmp $0x1,%eax | |
| jne 0x1c9 <main+336> | |
| lea -0x10(%rbp),%rax | |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x147 <main+206> | |
| mov %rax,%rdx | |
| lea -0xc(%rbp),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x159 <main+224> | |
| mov -0x10(%rbp),%eax | |
| mov %eax,%esi | |
| lea 0x0(%rip),%rdi | |

| | |
|---|---|
| callq 0x16a <main+241> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x179 <main+256> | |
| mov %rax,%rdx | |
| mov -0xc(%rbp),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x189 <main+272> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x198 <main+287> | |
| mov %rax,%rcx | |
| mov -0x10(%rbp),%edx | |
| mov -0xc(%rbp),%eax | |
| sub %eax,%edx | |
| mov %edx,%eax | |
| mov %eax,%esi | |
| mov %rcx,%rdi | |
| callq 0x1af <main+310> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x1c4 <main+331> | |
| jmpq 0x97 <main+30> | |
| mov -0x14(%rbp),%eax | Performs addition on user input and outputs the result in a formatted statement. |
| cmp $0x2,%eax | |
| jne 0x268 <main+495> | |
| lea -0x10(%rbp),%rax | |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x1e8 <main+367> | |
| mov %rax,%rdx | |
| lea -0xc(%rbp),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x1fa <main+385> | |
| mov -0x10(%rbp),%eax | |
| mov %eax,%esi | |
| lea 0x0(%rip),%rdi | |
| callq 0x20b <main+402> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x21a <main+417> | |
| mov %rax,%rdx | |
| mov -0xc(%rbp),%eax | |
| mov %eax,%esi | |

| | |
|---|---|
| mov %rdx,%rdi | |
| callq 0x22a <main+433> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x239 <main+448> | |
| mov %rax,%rcx | |
| mov -0x10(%rbp),%edx | |
| mov -0xc(%rbp),%eax | |
| add %edx,%eax | |
| mov %eax,%esi | |
| mov %rcx,%rdi | |
| callq 0x24e <main+469> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x263 <main+490> | |
| jmpq 0x97 <main+30> | |
| mov -0x14(%rbp),%eax | Performs division on user input and outputs the result in a formatted statement. |
| cmp $0x3,%eax | |
| jne 0x97 <main+30> | |
| lea -0x10(%rbp),%rax | |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x287 <main+526> | |
| mov %rax,%rdx | |
| lea -0xc(%rbp),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x299 <main+544> | |
| mov -0x10(%rbp),%eax | |
| mov %eax,%esi | |
| lea 0x0(%rip),%rdi | |
| callq 0x2aa <main+561> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x2b9 <main+576> | |
| mov %rax,%rdx | |
| mov -0xc(%rbp),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x2c9 <main+592> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x2d8 <main+607> | |
| mov %rax,%rcx | |
| mov -0x10(%rbp),%eax | |
| mov -0xc(%rbp),%esi | |

| | |
|---|---|
| cltd | |
| idiv   %esi | |
| mov    %eax,%esi | |
| mov    %rcx,%rdi | |
| callq  0x2ee <main+629> | |
| mov    %rax,%rdx | |
| mov    0x0(%rip),%rax | |
| mov    %rax,%rsi | |
| mov    %rdx,%rdi | |
| callq  0x303 <main+650> | |
| jmpq   0x97 <main+30> | |
| mov    $0x0,%eax | Resets value |
| mov    -0x8(%rbp),%rcx | Loops back to beginning |
| xor    %fs:0x28,%rcx | |
| je     0x321 <main+680> | |
| callq  0x321 <main+680> | Exits program |
| leaveq | |
| retq | |

| Blocks of Assembly Code | C++ Code |
|---|---|
| | int menuSelection = 0, num1, num2; |
| cmp $0x5,%eax<br>je 0x308 <main+655> | while (menuSelection < 5)) { |
| lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xb6 <main+61><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xc9 <main+80><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xdc <main+99><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xef <main+118> | DisplayMenu(); |
| lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x102 <main+137> | std::cin >> menuSelection; |
| lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x115 <main+156> | std::cin >> num1; |
| lea -0x14(%rbp),%rax<br>mov %rax,%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x128 <main+175> | std::cin >> num2; |
| mov -0x14(%rbp),%eax<br>cmp $0x1,%eax<br>jne 0x1c9 <main+336> | if (menuSelection == 1) { |
| | std::cout << num1 << " + " << num2 << " = " << num1 + num2 << std::endl; |

| | |
|---|---|
| lea -0x10(%rbp),%rax<br>mov %rax,%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x147 <main+206><br>mov %rax,%rdx<br>lea -0xc(%rbp),%rax<br>mov %rax,%rsi<br>mov %rdx,%rdi<br>callq 0x159 <main+224><br>mov -0x10(%rbp),%eax<br>mov %eax,%esi<br>lea 0x0(%rip),%rdi<br>callq 0x16a <main+241><br>lea 0x0(%rip),%rsi<br>mov %rax,%rdi<br>callq 0x179 <main+256><br>mov %rax,%rdx<br>mov -0xc(%rbp),%eax<br>mov %eax,%esi<br>mov %rdx,%rdi<br>callq 0x189 <main+272><br>lea 0x0(%rip),%rsi<br>mov %rax,%rdi<br>callq 0x198 <main+287><br>mov %rax,%rcx<br>mov -0x10(%rbp),%edx<br>mov -0xc(%rbp),%eax<br>sub %eax,%edx<br>mov %edx,%eax<br>mov %eax,%esi<br>mov %rcx,%rdi<br>callq 0x1af <main+310><br>mov %rax,%rdx<br>mov 0x0(%rip),%rax<br>mov %rax,%rsi<br>mov %rdx,%rdi<br>callq 0x1c4 <main+331><br>jmpq 0x97 <main+30> | } |
| mov -0x14(%rbp),%eax<br>cmp $0x2,%eax<br>jne 0x268 <main+495> | else if (menuSelection == 2) { |
| | std::cout << num1 << " - " << num2 << " = " << num1 - num2 << std::endl; |

| | |
|---|---|
| lea -0x10(%rbp),%rax | } |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x1e8 <main+367> | |
| mov %rax,%rdx | |
| lea -0xc(%rbp),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x1fa <main+385> | |
| mov -0x10(%rbp),%eax | |
| mov %eax,%esi | |
| lea 0x0(%rip),%rdi | |
| callq 0x20b <main+402> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x21a <main+417> | |
| mov %rax,%rdx | |
| mov -0xc(%rbp),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x22a <main+433> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x239 <main+448> | |
| mov %rax,%rcx | |
| mov -0x10(%rbp),%edx | |
| mov -0xc(%rbp),%eax | |
| add %edx,%eax | |
| mov %eax,%esi | |
| mov %rcx,%rdi | |
| callq 0x24e <main+469> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x263 <main+490> | |
| jmpq 0x97 <main+30> | |
| mov -0x14(%rbp),%eax | else if (menuSelection == 3) { |
| cmp $0x3,%eax | std::cout << num1 << " * " << num2 << " = " << num1 * num2 << std::endl; |
| jne 0x97 <main+30> | |

| | |
|---|---|
| lea -0x10(%rbp),%rax<br>mov %rax,%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x287 <main+526><br>mov %rax,%rdx<br>lea -0xc(%rbp),%rax<br>mov %rax,%rsi<br>mov %rdx,%rdi<br>callq 0x299 <main+544><br>mov -0x10(%rbp),%eax<br>mov %eax,%esi<br>lea 0x0(%rip),%rdi<br>callq 0x2aa <main+561><br>lea 0x0(%rip),%rsi<br>mov %rax,%rdi<br>callq 0x2b9 <main+576><br>mov %rax,%rdx<br>mov -0xc(%rbp),%eax<br>mov %eax,%esi<br>mov %rdx,%rdi<br>callq 0x2c9 <main+592><br>lea 0x0(%rip),%rsi<br>mov %rax,%rdi<br>callq 0x2d8 <main+607><br>mov %rax,%rcx<br>mov -0x10(%rbp),%eax<br>mov -0xc(%rbp),%esi<br>cltd<br>idiv %esi<br>mov %eax,%esi<br>mov %rcx,%rdi<br>callq 0x2ee <main+629><br>mov %rax,%rdx<br>mov 0x0(%rip),%rax<br>mov %rax,%rsi<br>mov %rdx,%rdi<br>callq 0x303 <main+650><br>jmpq 0x97 <main+30> | } |
| | } |