

Phase 1: The answer is hidden inside the binary file code and to get it I needed to use the strings command.

Phase 2: First I made sure that indeed 6 integers are needed so that the bomb does not explode. Therefore I inputted 3 random strings and observed the assembly code to see

if the assembly code would eventually lead to the explode function. Then I inputted 6 ints as inputs to observe the comparisons that the assembly code makes. From the

comparisons in the assembly we see that the first number needs to be 0 and the second needs to be 1. From the rest of the assembly code we can conclude that this is a

Fibonacci Sequence. This is because in the assembly a loop occurs where the new value is the sum of the previous 2 values of the input.

Phase 3: First I determined the input format which is given in one of the assembly lines as %d %d. Then by the comparison statement I saw that the first number has to be int and is

less than or equal to 7. Then I saw that the first value entered determines the correct value for the second number. And in order to find the second number I had to do couple

of hexadecimal additions in order to obtain various addresses that we needed in order to jump to them as a next step.

Phase 4: We again find the input format(%d %d) and then see that we enter a recursive function. After computation, I found out that our inputs were switched as well. After computation

and following the disassembly, I concluded that the right input is 132 4