PHASE 2 BINARY BOMB

JOHN THOMAS
P2CSN17017

Phase 2 reads six numbers from the input string using sscanf. If there are
less than 6 numbers, the bomb explodes. Then, the phase checks if these six
is defused, else the bomb explodes.

1. There is a function **sub_8048B48** ,which start the phase 2.
   Inside the loop we see that there is two another function calls i.e, **sub_8048FD8** and **sub_8048B6E**

2. We analyse each function seperately,
when we call **sub_8048FD8**, we enter into that function,we see that the stack frame is created and coming down we see a call _sscanf used to input data, which pushes the offset which contains %d printed 6 times.So we can say that the input would be 6 digits.

3. When we analyse the function, we see at location **08049007** there is cmp eax , 5 i.e, comparing the 6 values, If the cmp is correct with the entered number of digits, then it performs a jumb greater to the location **loc_8049011**, else it will call the function **sub_80494FC(explode function)** which prints that the bomb has exploded.

4. Now again coming back into **sub_8048FD8** we see a comparison [ebp+var_18],1.which is a stack address which compares the first value given by user with 1.It checks whether both values are equal. If both values are equal ,then it jumbs to location **loc_8048B6E,** else jumbs to **explode function** which blasts the bomb.

5. In **loc_8048B6E,** it will mov value 1 into ebx
   Next it goes into **loc_8048B76,** where eax value is incremented from 1 to 2 and multiplication operation takes place **eax , [esi + ebx*4 -4]** where it calculates the memory location of esi and takes the value in the memory location and compares it with eax.If both the values are equal, it jumbs to **loc_8048B88.**The value 1 is stored in eax.

6.Coming inside **loc_8048B88 ,** it again increments the value of ebx and compares the value of ebx with 5. If the condition is met then, jump to **loc_8048B76.**

```
Lea   eax, [ebx+1]
imul   eax, [esi+ebx*4-4]
```

Again the ebx value is incremented to 1 and checks the loop , multiply and compare the value of eax with esi+4 value.It will be stored in eax.The value 2 is stored in eax.

Again ebx value is incremented to 2 and checks the loop and multiply and compare the value of eax with esi+8 value. It will be stored in eax.The value 6 is stored in eax.

Again ebx value is incremented to 3 and checks the loop and multiply and compare the value of eax with esi+12 value.It will be stored in eax.The value 24 is stored in eax.

Again ebx value is incremented to 4 and checks the loop and multiply and compare the value of eax with esi+16 value. It will be stored in eax.The value 120 is stored in eax.

Again ebx value is incremented to 5 and checks the loop, multiply and compare the value of eax with esi+20 value.It will be stored in eax.The value 720 is stored in eax.
Now the Last user input is compared with the eax .

So when,
ebx=0 ; eax = 1
ebx=1 ; eax = 2
ebx=2 ; eax = 6
ebx=3 ; eax = 24
ebx=4 ; eax = 120
ebx=5 ; eax = 720

Now again ebx value is incremented to 6.Now its compared with 5 ,but 6 >5
So, it stops there and exits out of the loop through return function and calls the function **sub_8084B48**

So the output is **1, 2, 6, 24, 120, 720**