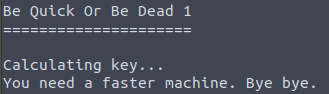
Reverse 5: Be Quick Or Be Dead 1

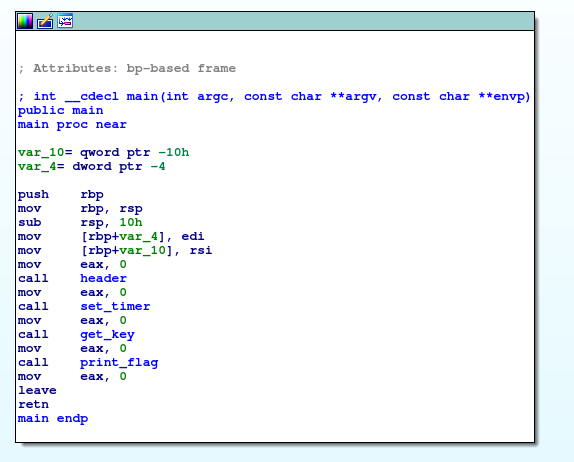
**Issue:** IDA doesn’t allow me to modify as I want the instructions. For example, I would like to increase the timer to 8 (as proposed in the solution) but, however, I can’t and I receive an error. Is it due to the fact that I do not use the pro version?

The description says “my machine is too slow for executing the program and reach the flag”. At this point, I have no idea of what it means.

We can execute the program.



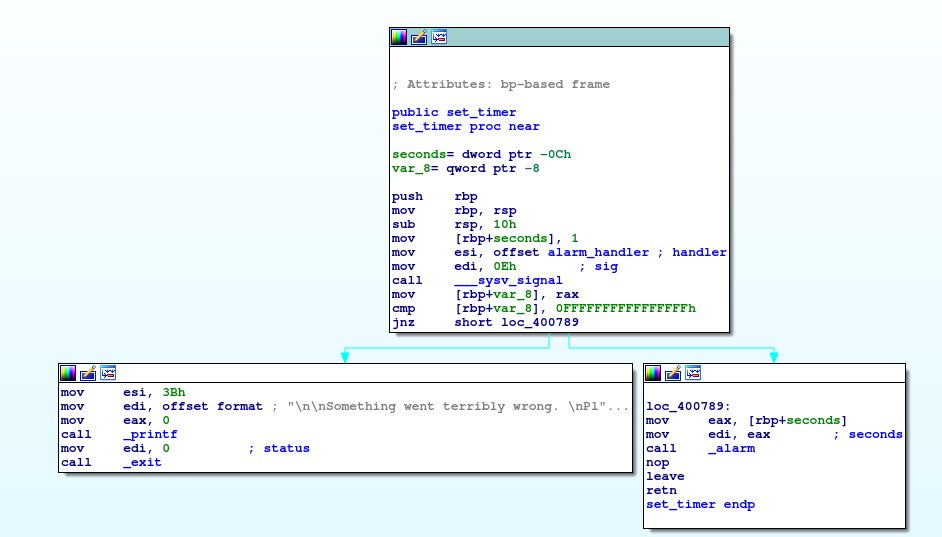
We cannot neither insert / interact with the program. Based on our “big” set of tools (objdump - strings - IDA), a proper guess is that we need to “help” the flow execution of the program. Let’s open the file with IDA.



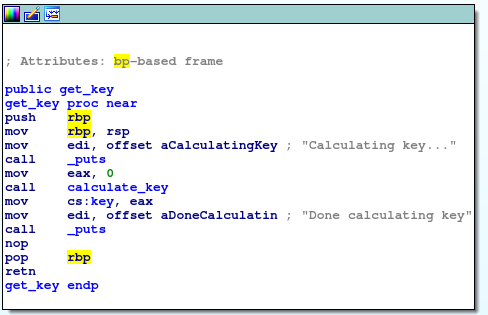
The program seems “naive”. Some functions are called, such as “set timer”. Two functions seem interesting:

* Set timer
* Get key

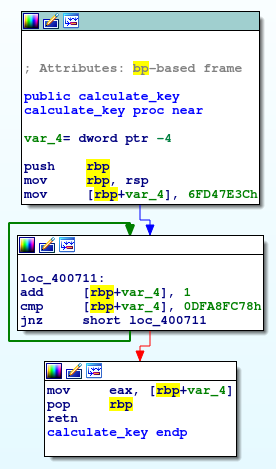
Double click on “set\_timer” and we’ll see its assembly:



This function set an alarm to 1 second and when it goes to 0 the program’s execution will terminate. Let’s move to *get key*:



This function calculates a key with *calculate\_key* … nothing more. Let’s study also *calculate\_key*.



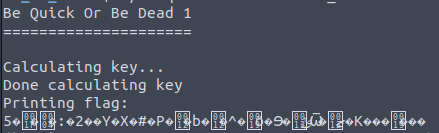
The second block is a clear *while* loop, where a variable is incremented until a certain amount is reached. If we see also the function *print\_flag*, the last one, no *cmp* or additional *loops* are defined.

Well, the description talk about issues with the time of execution. A possible explanation is that the alarm stops the program’s execution before the print\_flag is called, due to the loop.

With IDA we can modify the program and try to reach the flag … for example we can:

* Increase the amount of seconds of the alarm;
* Generate the key immediately, by changing the *jnz* with *jz*;
* There are tons of other possibilities.

I’ll try by replacing the *jnz* with *jz* in the loop.



We reached the flag .. but it is unreadable! Maybe we need to reach the proper key in order to have the flag .. it makes sense, since *print\_flag* uses *decrypt\_flag* (we know that the encryption / decryption algorithms must use correct key).

I can try with my first hypothesis: increase the seconds of the alarm; let’s try with 60 seconds (hopefully it is enough).