

Intro to Reverse Engineering

What does Reverse Engineering mean?

It means **analyzing** for the purpose to **understand** how it works.

This process helps us **finding bugs and vulnerabilities** that we can exploit to find what we need, sometimes even directly the flag.

Reverse engineering might allow us to find **useful data** (for example passwords, DLC keys, seeds, etc.) or **unintended behaviours**. In some cases we can even **tamper** some parts of the code.

(like buffing our speed in a **videogame** or even implementing fly hacks as we'll see in the next lesson)

What are we “reversing”?

- Humans and machines do not speak the same language.
- Programs need to be compiled into binaries, sequences of zeros and ones that computers can understand.
- Our goal is to reverse this process to understand what the executable is doing under the hood.

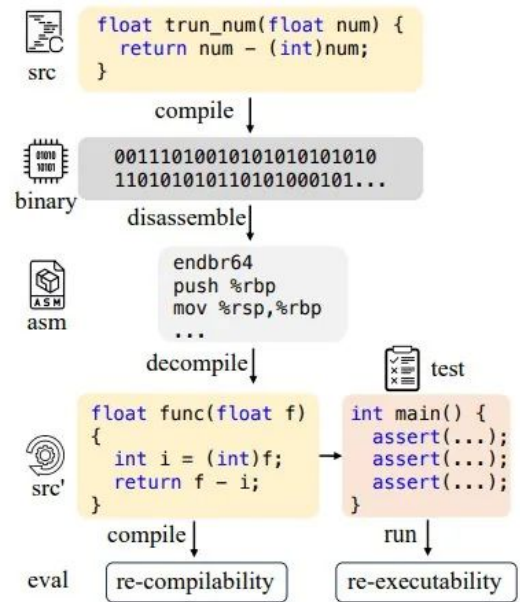
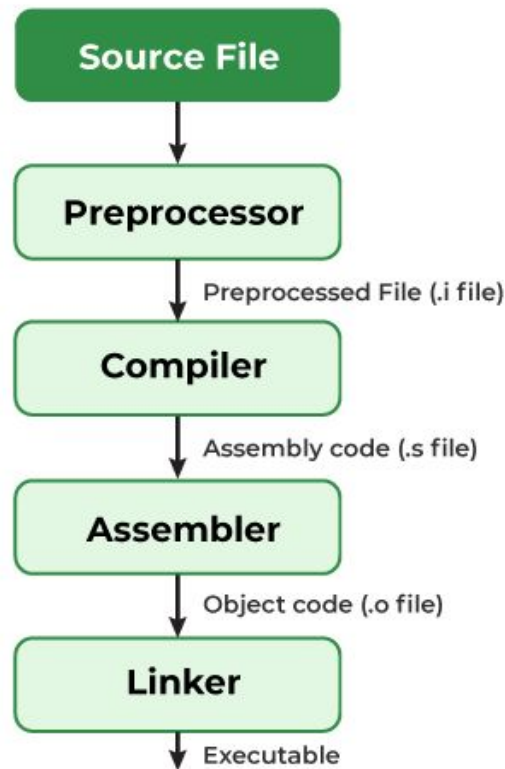


Figure 1: Pipeline to evaluate the decompilation.

From code to executable

In order to be used in **different** machines, a program must be compiled differently according to the architecture and OS.

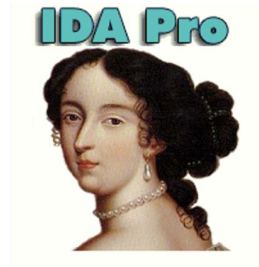
Compilation though is a **lossy** process



Decompiling the executable

Decompiling means trying to obtain a **higher-level** version (the language the original program was written in) of the source code for a better understanding. Most of the time the code we obtain this way is **obfuscated**, nonetheless it is useful to have a **more comprehensible** version than just assembly.

There are various decompilers we might use to analyze our executable, like **Ghidra**, **IDA** or **Binary Ninja**.



Ghidra

- Ghidra is a free and open source **reverse engineering** tool developed by the NSA.
- Lots of plugins and utilities
- Really, it is free.

