Introduction to Cybersecurity

Lecture #4: Reverse engineering II

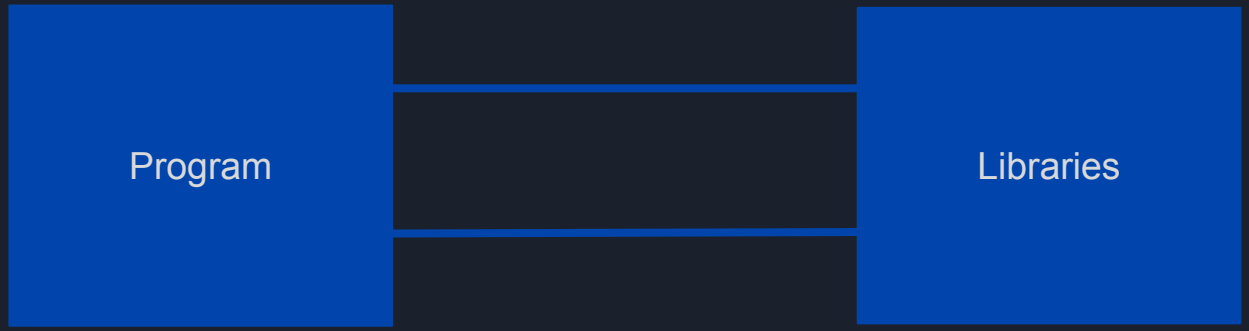
Anton Semenkin

Reverse engineering

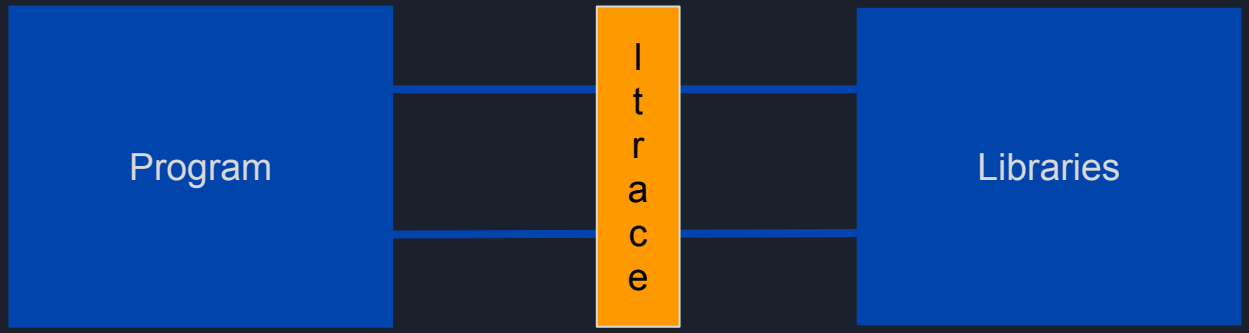
Reverse levels

- Source code analysis
- Tracing
- Static analysis
- Binary patching

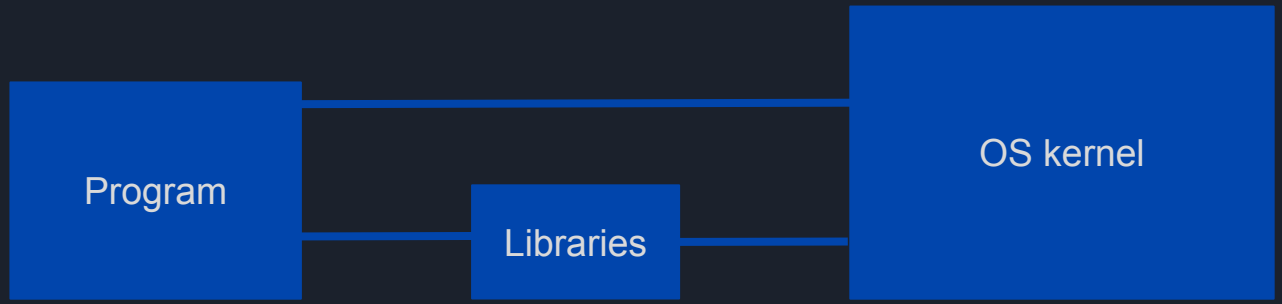
Tracing: functions interception



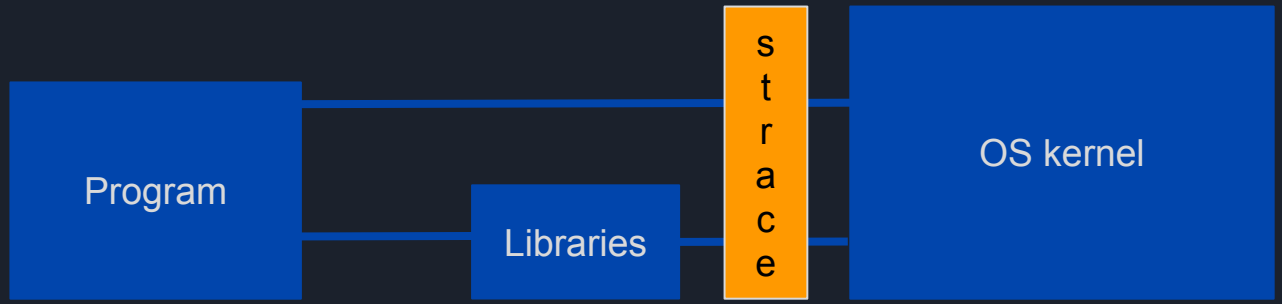
Tracing: functions interception



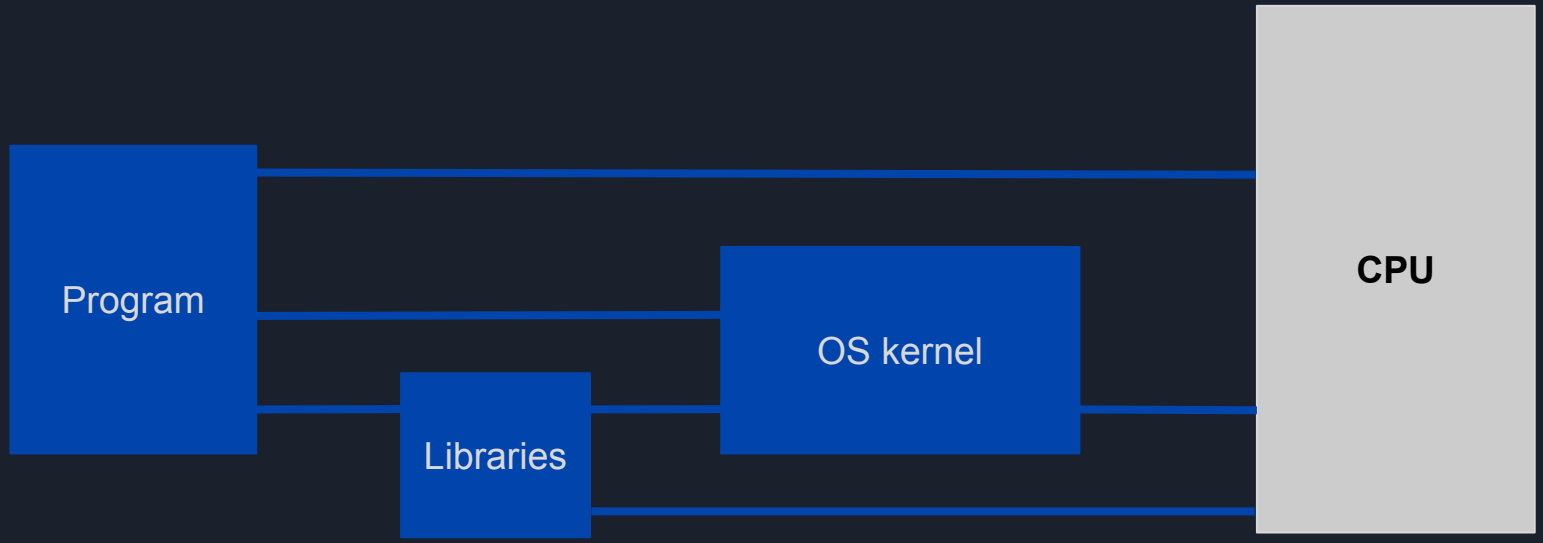
Tracing programs: syscalls interception



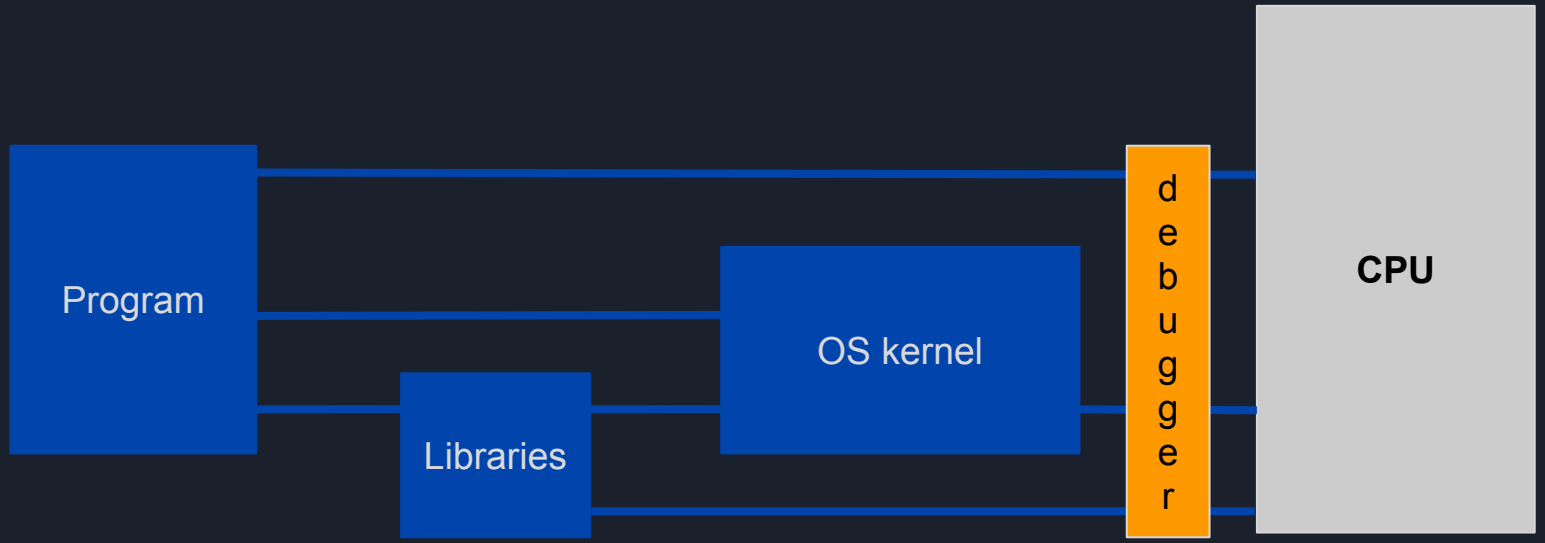
Tracing programs: syscalls interception



Tracing program: process memory access



Tracing program: process memory access



Tools overview

- Binutils
  - strings
  - objdump
  - gprof
  - readelf
  - other
- Tracing of all types
  - strace
  - ltrace
  - gdb

https://www.gnu.org/software/binutils/

Tools overview

- Binutils
  - strings
  - objdump
  - gprof
  - readelf
  - other
- Tracing of all types
  - strace
  - ltrace
  - gdb

USE MAN

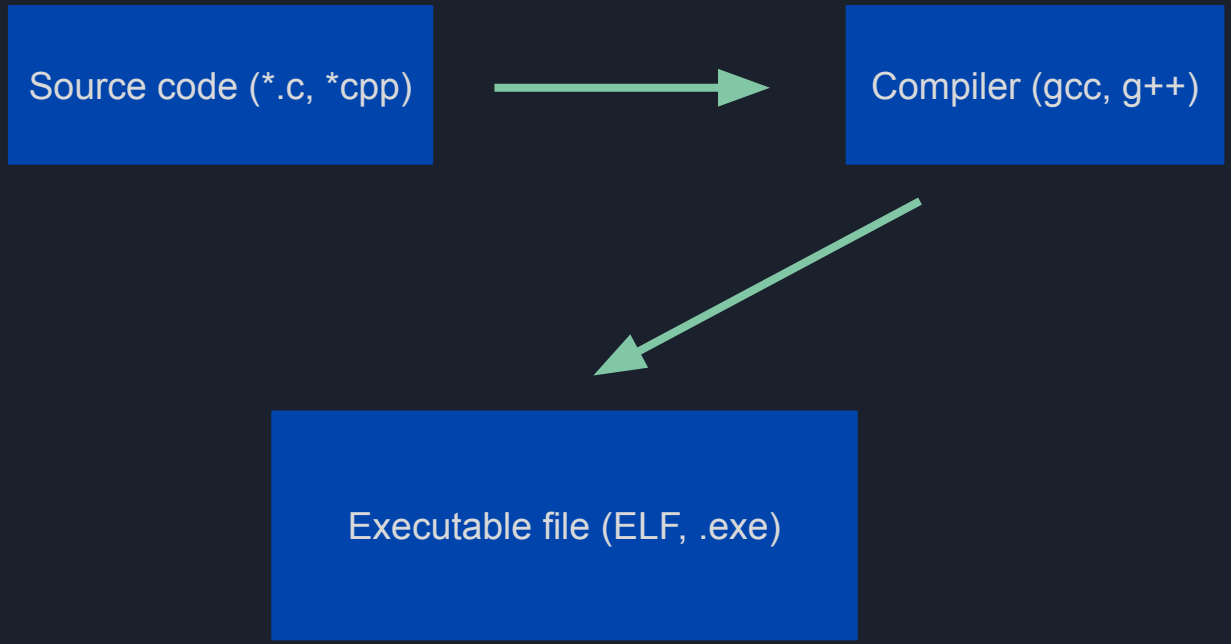
https://www.gnu.org/software/binutils/

Demo

Programming languages levels

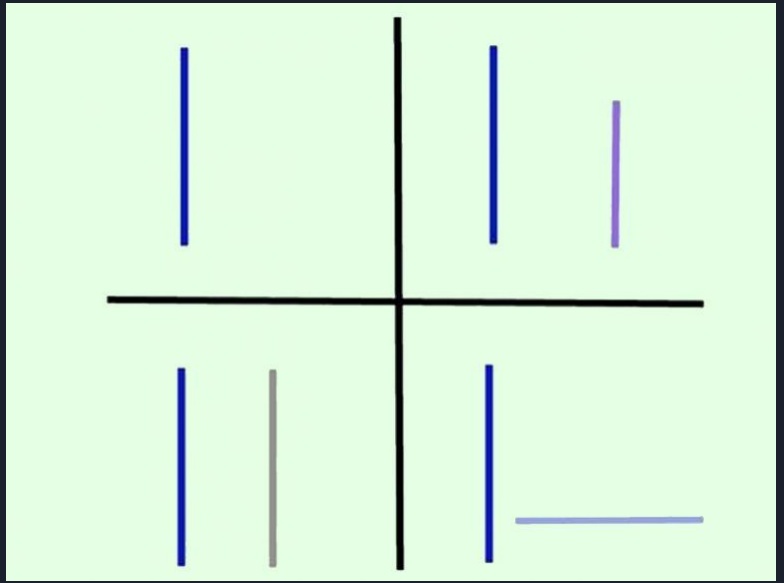
- Python, Perl, Ruby, C#, Java
- C, C++
- Assembly
- Machine code

Compilation process



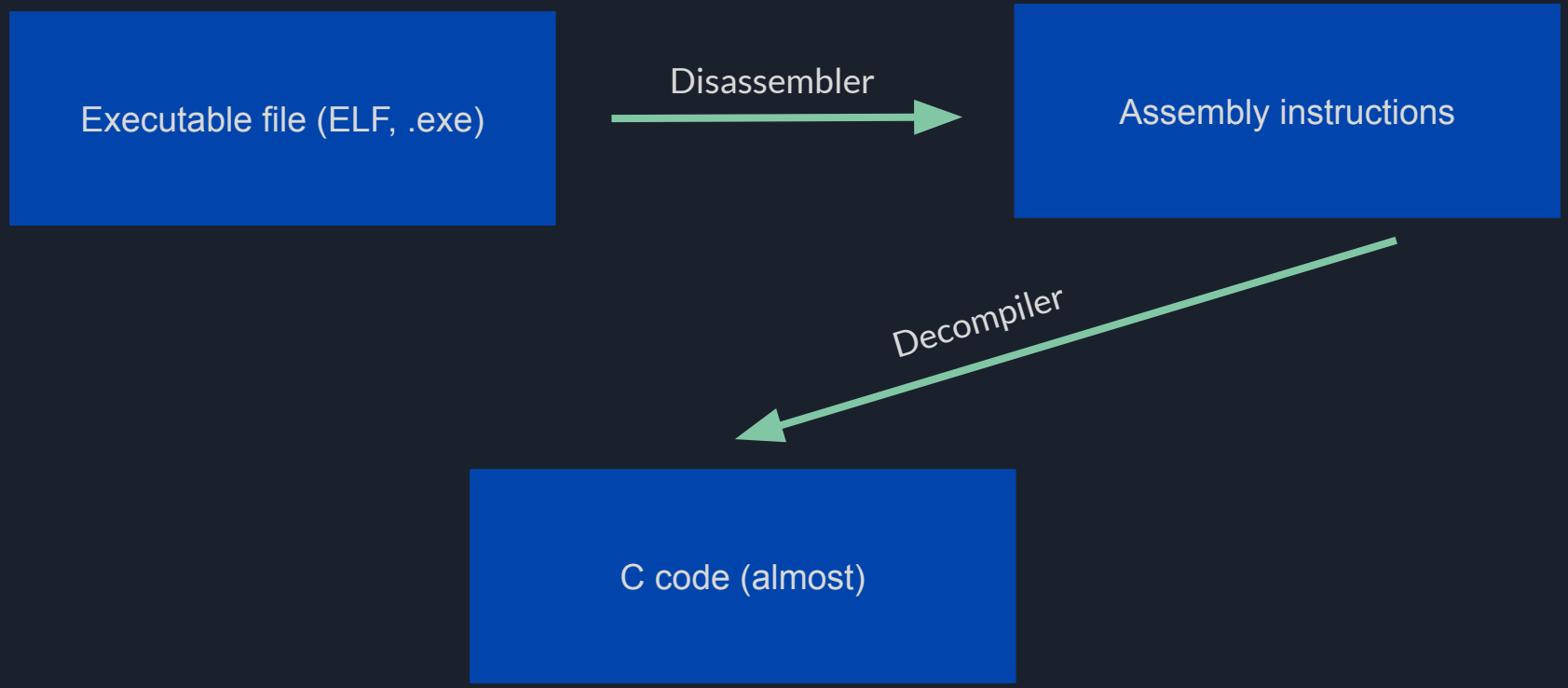
Compilation process

- Information loss
  - variables names
  - comments
  - structs
  - classes
  - objects



Demo

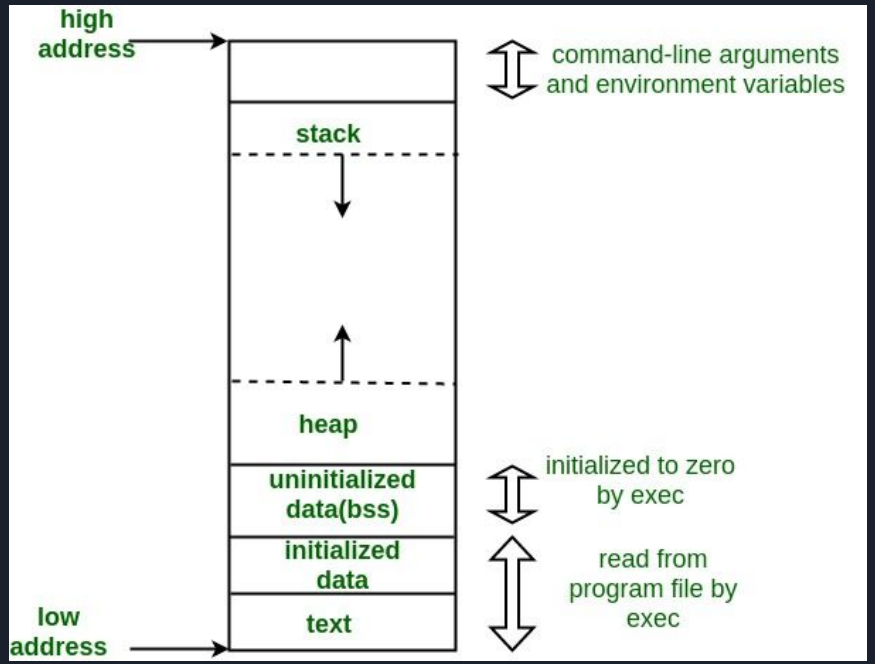
Reverse engineering



Demo

Memory layout

- Static
  - .bss, .data
- Dynamic
  - heap
- Automatic
  - stack



https://www.geeksforgeeks.org/memory-layout-of-c-program/

Variable size

- db - 1 byte
  - char
- dw - 2 bytes
  - short
- dd - 4 bytes
  - int, float
- dq - 8 bytes
  - long long, double

Java demo