CHECK-IN

KUISC - REVERSE ENGINEERING

WEDNESDAY, OCTOBER 14 AT 6:00PM

https://cglink.me/2rx/c3778801013181922

Reverse Engineering

Meeting 7

Announcements

- NSA Codebreaker still active



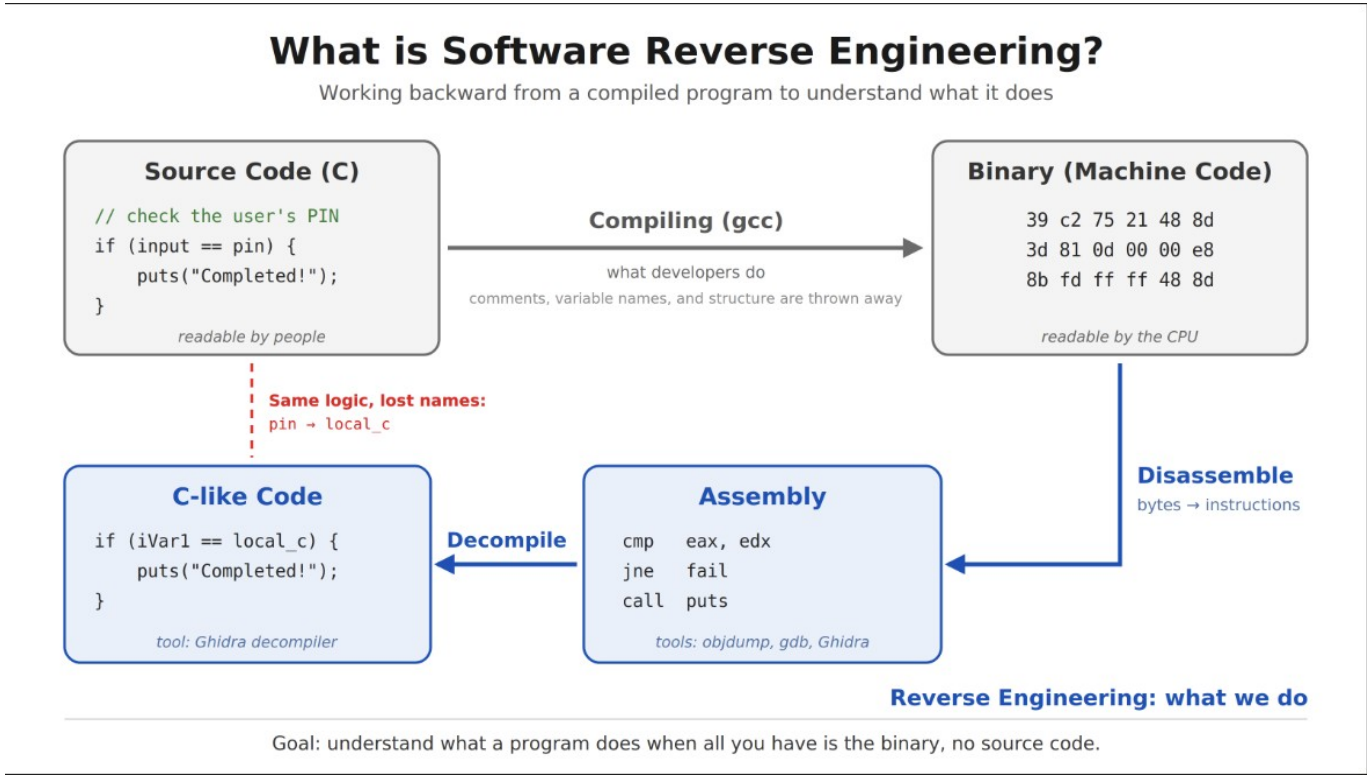
Division	Rank	School	Score	Task 0	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
D-3	25	University of Kansas	75,422	7	8	4	2	0	0	0	0	0

What is Reverse Engineering?

- Analyzing a compiled program (a binary) without its source code

- Some information about the source code can be lost when a program is compiled into a binary, the goal of reverse engineering is to piece together the missing information

- Used for malware analysis, CTF competitions, finding vulnerabilities, and old or undocumented software



Types of Binary Analysis

- **Static Analysis – Examining a program without running it**
  - Pros: Safe, can see the whole program
  - Cons: Can't see the value of the variables at different points of the program

- **Dynamic Analysis - Running a program and watching it**
  - Pros: Can watch values change in real time
  - Cons: Running unknown code is risky, can only see the path that is run

Static Analysis Tools

- file – what kind of binary it is
- strings – prints every readable piece of text in the program
- objdump -d – dumps the disassembly into the terminal
- nm/readelf – list function names, variables, and sections
- Ghidra – Disassembler and decompiler created by the NSA, turns machine code back into readable C like code

Dynamic Analysis Tools

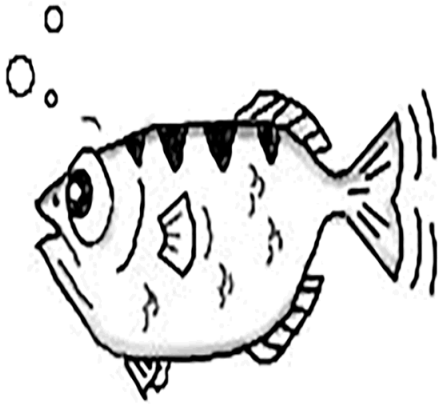
- Just running the program - ./program

- gdb – the GNU debugger. It can pause a program midway through, read and change registers and memory

- strace/ltrace - show the system calls and library calls a program makes while running

- Always run unknown binaries on a VM

Demo



GDB

The GNU Project

Debugger





CHALLENGE TIME

Reverse Engineering Lab

Situation – Reverse engineer the binaries to find information about the programs and hidden flags

User: kali
Pass: toor

First to complete the challenge gets a brownie!

