Back Doors, C

CS 2130: Computer Systems and Organization 1

October 17, 2022

Announcements

- Homework 5 due Wednesday 10/19 at 11pm

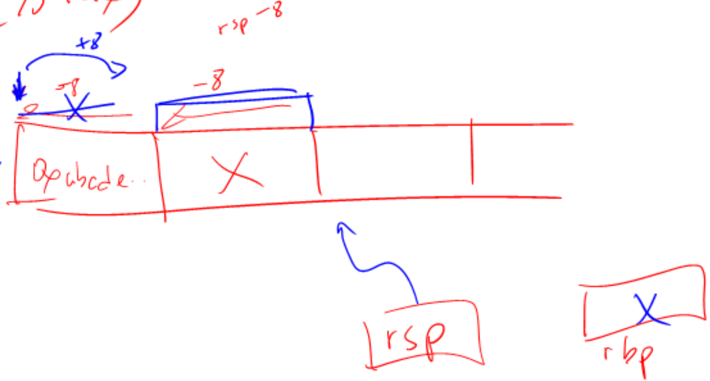
Quiz Review

4,5,6,1

offset (%reg1, %reg2, multiple)

8 (%rax)

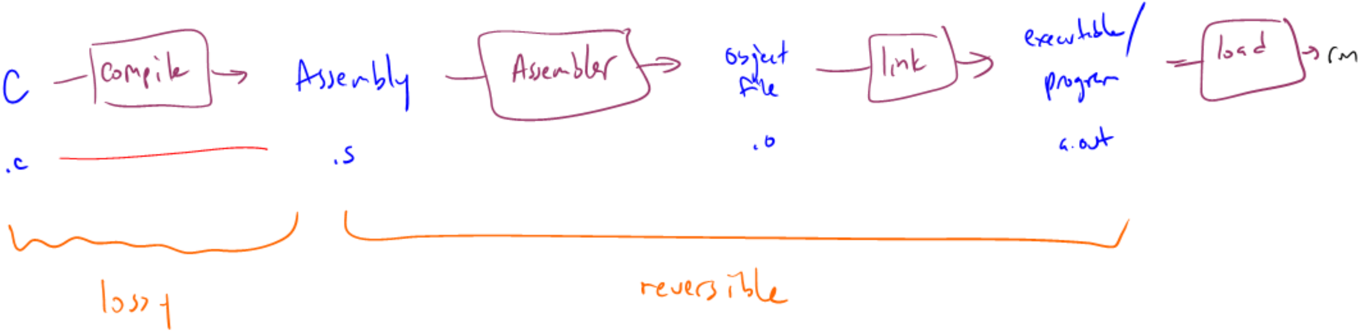
movq 8(%rsp), %rax



Compilation Pipeline

Turning our code into something that runs

- **Pipeline** - a sequence of steps in which each builds off the last



Most Common Instructions

- `mov` - =
- `lea` - load effective address
- `call` - push PC and jump to address
- `add` - +=
- `cmp` - set flags as if performing subtract
- `jmp` - unconditional jump
- `test` - set flags as if performing &
- `je` - jump iff flags indicate == 0
- `pop` - pop value from stack
- `push` - push value onto stack
- `ret` - pop PC from the stack

A short aside…

Time to take over the world!

Backdoors

Backdoor: secret way in to do new *unexpected* things

- Get around the normal barriers of behavior
- Ex: a way in to allow me to take complete control of your computer

Exploit - a way to use a vulnerability or backdoor that has been created

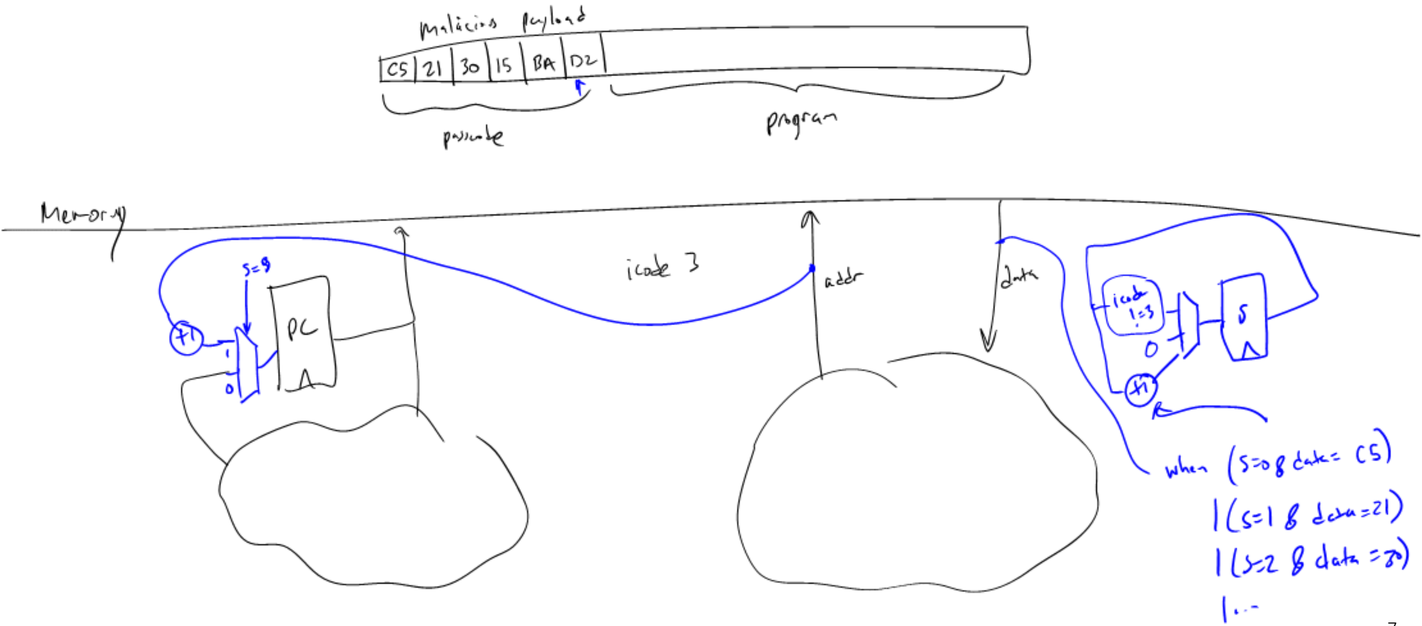
- Our exploit today: a **malicious payload**
    - A passcode and program
    - If it ever gets in memory, run my program regardless of what you want to do

Our Hardware Backdoor

Our backdoor will have 2 components

- Passcode: need to recognize when we see the passcode
- Program: do something bad when I see the passcode

Our Hardware Backdoor



Our Hardware Backdoor

Will you notice this on your chip?

Our Hardware Backdoor

Will you notice this on your chip?

- Modern chips have **billions** of transistors
- We’re talking adding a few hundred transistors

Our Hardware Backdoor

Will you notice this on your chip?

- Modern chips have **billions** of transistors
- We're talking adding a few hundred transistors
- *Maybe with a microscope? But you'd need to know where to look!*

Our Hardware Backdoor

Have you heard about something like this before?