Defenses

against **low-level attacks**

Stepping back

What do these attacks have in common?

1. The **attacker** is able to **control some data** that is used by the program

2. The use of that data **permits unintentional access to some memory area** in the program
   - past a buffer
   - to arbitrary positions on the stack

Outline

- **Memory safety** and **type safety**
  - Properties that, if satisfied, ensure an application is immune to memory attacks
- Automatic defenses
  - **Stack canaries**
  - Address space layout randomization (**ASLR**)
- Return-oriented programming (**ROP**) attack
  - How Control Flow Integrity (**CFI**) can defeat it
- **Secure coding**