Chapter 2. Introduction to Software Vulnerability

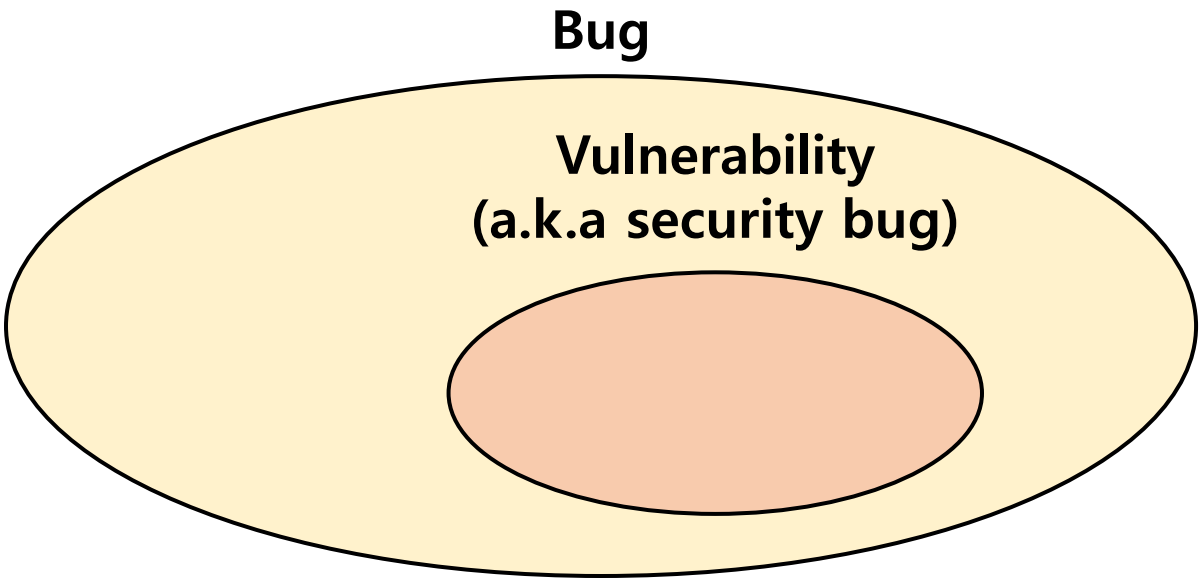
Prof. Jaeseung Choi

Dept. of Computer Science and Engineering

Sogang University

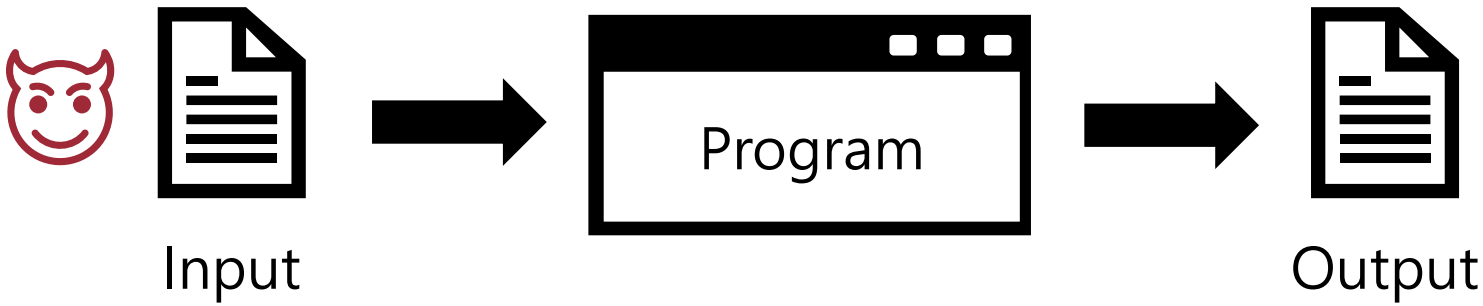
Software Bug and Vulnerability

- **Bug is an error in program that makes it malfunction**
  - Ex) Compute wrong outputs for corner case inputs
- **Vulnerability is a bug that causes security issues**
  - What kind of security issues? Review the previous slide
  - In some cases, the boundary can be ambiguous



How does a bug occur?

- **Usually, programmers implicitly assume that users will provide plain and benign inputs to the program**
- **However, hackers provide creative and malicious inputs that programmers *did not expect***



Thinking like an adversary is important for security

Example #1

- **Consider the python code below for *bank application***
  - Takes in the amount of money you want to transfer
  - Your balance and the recipient's balance will be updated

```
my_balance = 1000
def send(recipient):
    print("How much do you want to send?")
    val = read_int()
    if (val <= my_balance):
        my_balance = my_balance – val
        ... # Increase the balance of recipient
```

Input : **100**

my_balance: **1000 -> 900**

What can go wrong with this code?

Example #2

- **Consider the python code below for web service**
  - Creates a log directory for each user connected to the server
  - `os.system()` executes a command string in the shell

```
def service():
    username = read_from_packet()
    logdir = "./log/" + username
    cmdline = "mkdir %s" % logdir
    os.system(cmdline)
```

Input : **"jason"** → Command : **"mkdir ./log/jason"**

What can go wrong with this code?

Example #3

- **Next, consider the following C code**
  - Reads in a string input and prints it back

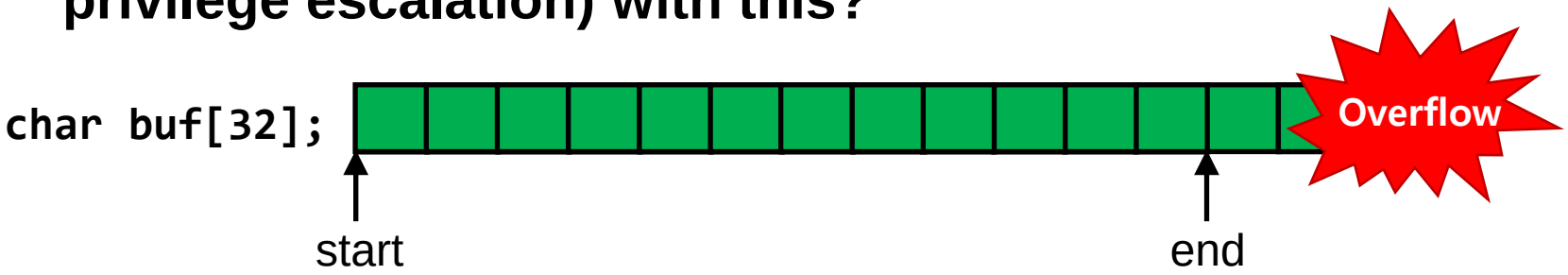
```
int main(void) {
    char buf[32];
    printf("Input your name: ");
    scanf("%s", buf);
    printf("Your name: %s\n", buf);
    return 0;
}
```

Input : **"Jason"** → Printed output: **"Your name: Jason"**

What can go wrong with this code?

Buffer Overflow & Memory Corruption

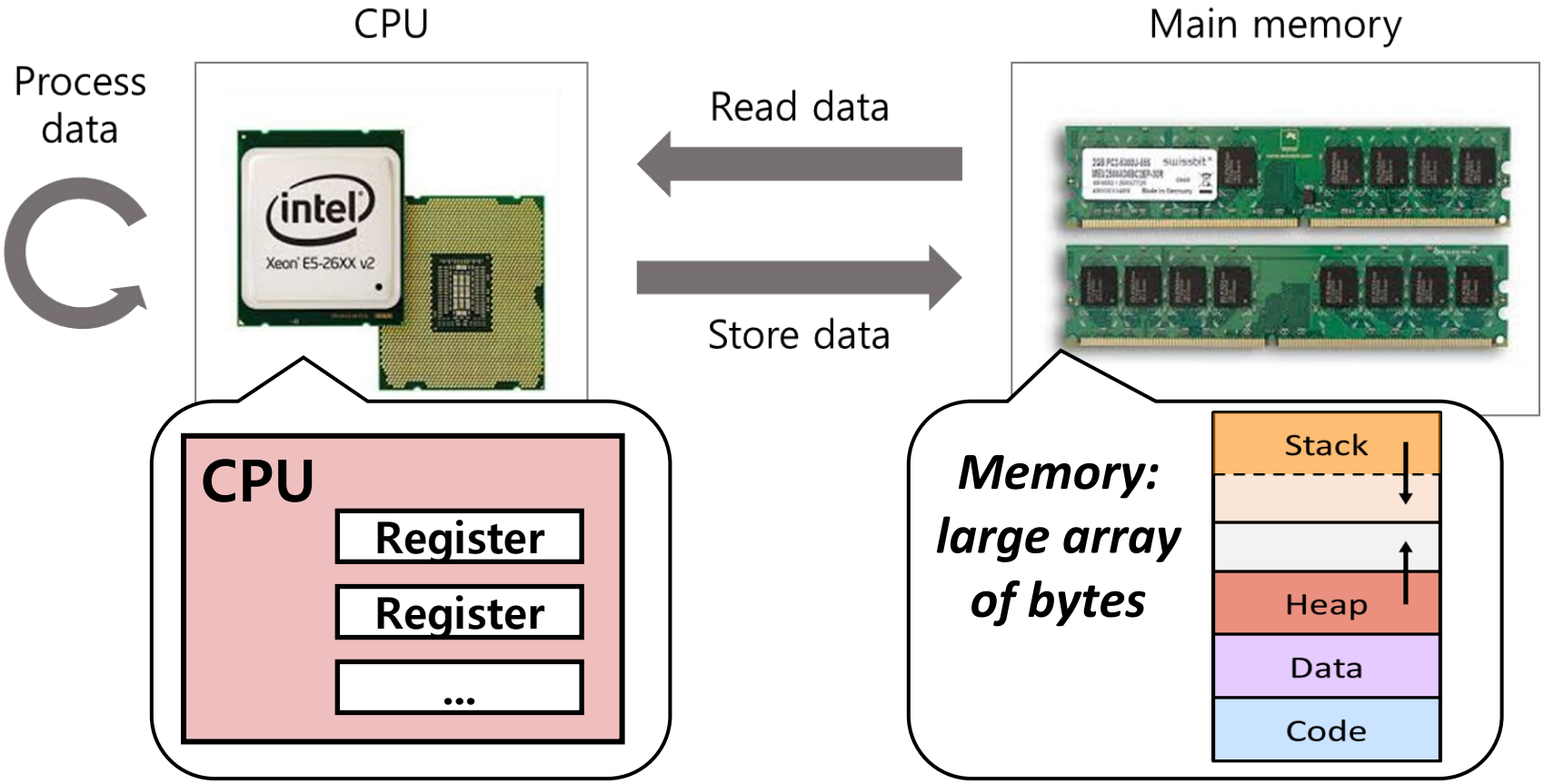
- **C has no automatic check on array index and boundary**
  - Allows writing **past the end of an array**
  - This is call buffer overflow, or **BOF** in short
  - Such write will **corrupt** other variables and data in the memory
- **Q. What kind of data will be corrupted exactly?**
- **Q. How can a hacker do bad things (e.g., code execution, privilege escalation) with this?**



To answer these, we must learn assembly

Low-level Internals

- **Assembly code controls computer's low-level operation**
  - The behavior of CPU and memory in your computer



In the next chapter, we will review Intel x86-64 assembly