Chapter 11. Course Review

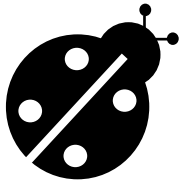
Prof. Jaeseung Choi

Dept. of Computer Science and Engineering

Sogang University

This Course: Software Security

- **In this course, we focused on *software security***
  - What kind of software **vulnerabilities** exist
  - How hackers can **exploit** those vulnerabilities
  - How to **prevent** hackers from exploiting those vulnerabilities
  - How to **detect** software vulnerabilities automatically (although we didn't have time to cover this, unfortunately)



Vulnerability

Exploitation

Mitigation

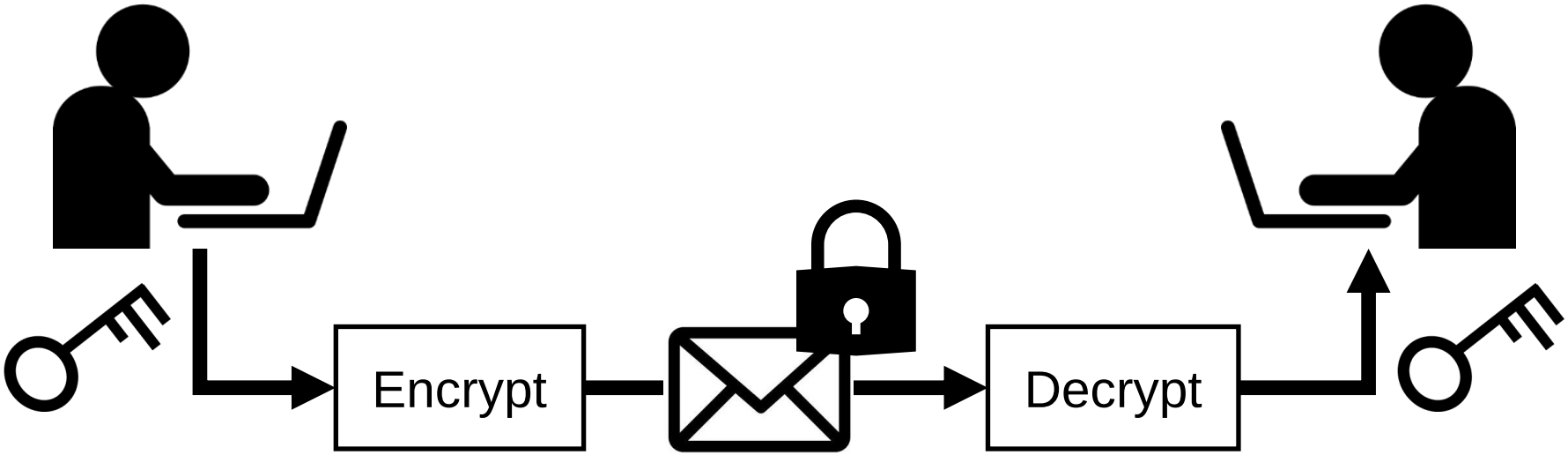
Detection

Other Topics in Security

- **There are many other topics in information security**
  - Cryptography
  - Network security
  - Hardware security
  - AI Security
  - … and many more
- **Let's skim through these topics briefly, before we review the topics we covered in this course**

Cryptography

- **Important tool for secure exchange of message**
  - **Confidentiality**: Adversary cannot know the message's content
  - **Integrity**: Ensures that message is not tampered by someone
- **Message (plaintext) is encrypted / decrypted with a key**
  - **Symmetric key** vs. **Asymmetric key** (public key cryptography)

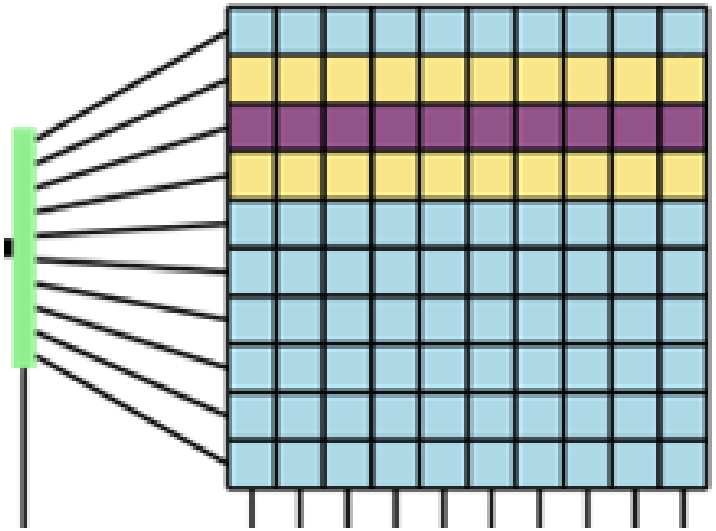


Network Security

- **How to protect the system and resources from various attackers at network level**
  - Sometimes, attacks become possible by the nature of protocols
- **Sniffing:** *eavesdropping* the packets in the network
  - Ex) In Ethernet protocol, segment shares the medium (wire)
- **Spoofing:** impersonating as someone else by manipulating network packet data
  - Ex) IP spoofing, ARP spoofing, DNS spoofing, …
- **(Distributed) Denial-of-service attacks**
  - Ex) SYN flooding, DNS amplification …

Hardware Security

- **Vulnerabilities in hardware (memory/CPU) design can also put the computer system at risk!**
  - Ex) **Rowhammer** attack: repeated access on DRAM memory's row can flip the bits of adjacent rows (electric disturbance)
  - Ex) **Spectre** and **Meltdown** attack: exploits speculative execution feature of modern CPUs

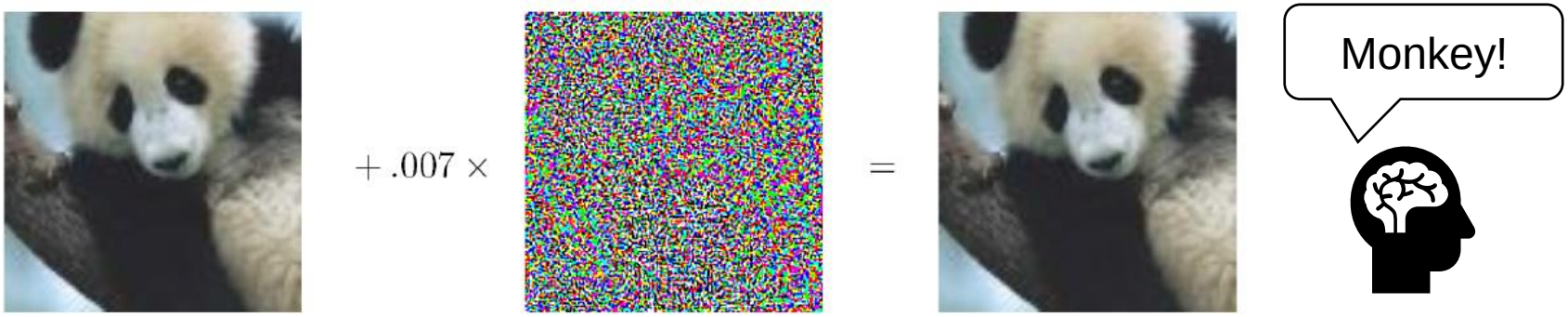


(High level illustration of Rowhammer attack)

Image from https://en.wikipedia.org/wiki/Row_hammer
https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

AI Security

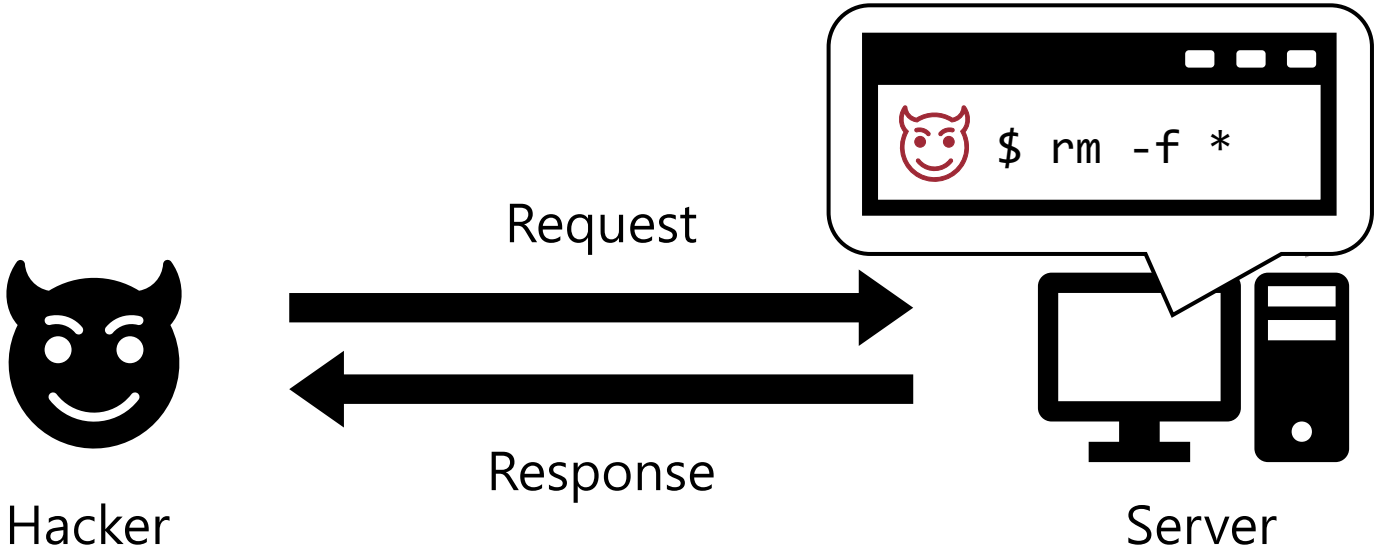
- **First of all, let me emphasize that Security for AI and AI for Security are two different things**
- **Security for AI:** how to the AI system itself
  - **Adversarial attacks**: making AI models to malfunction by providing maliciously crafted input
  - Privacy of user data during learning
- **AI for security:** using AI to solve problems in security
  - Ex) Use AI for intrusion detection or finding SW vulnerabilities (?)



Lastly, let’s review the topics we covered in SW security!

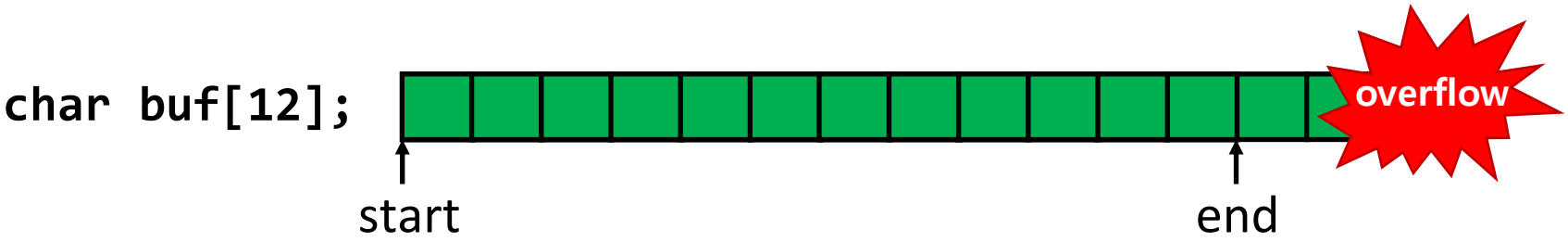
Hacking & Security

- **Various vulnerabilities can occur in computer systems**
  - Hackers can exploit them and pose serious threats
  - We will learn these attacks and the defense against them
- **Ex) Assume that your computer is running a service**
  - What if the service has a vulnerability?



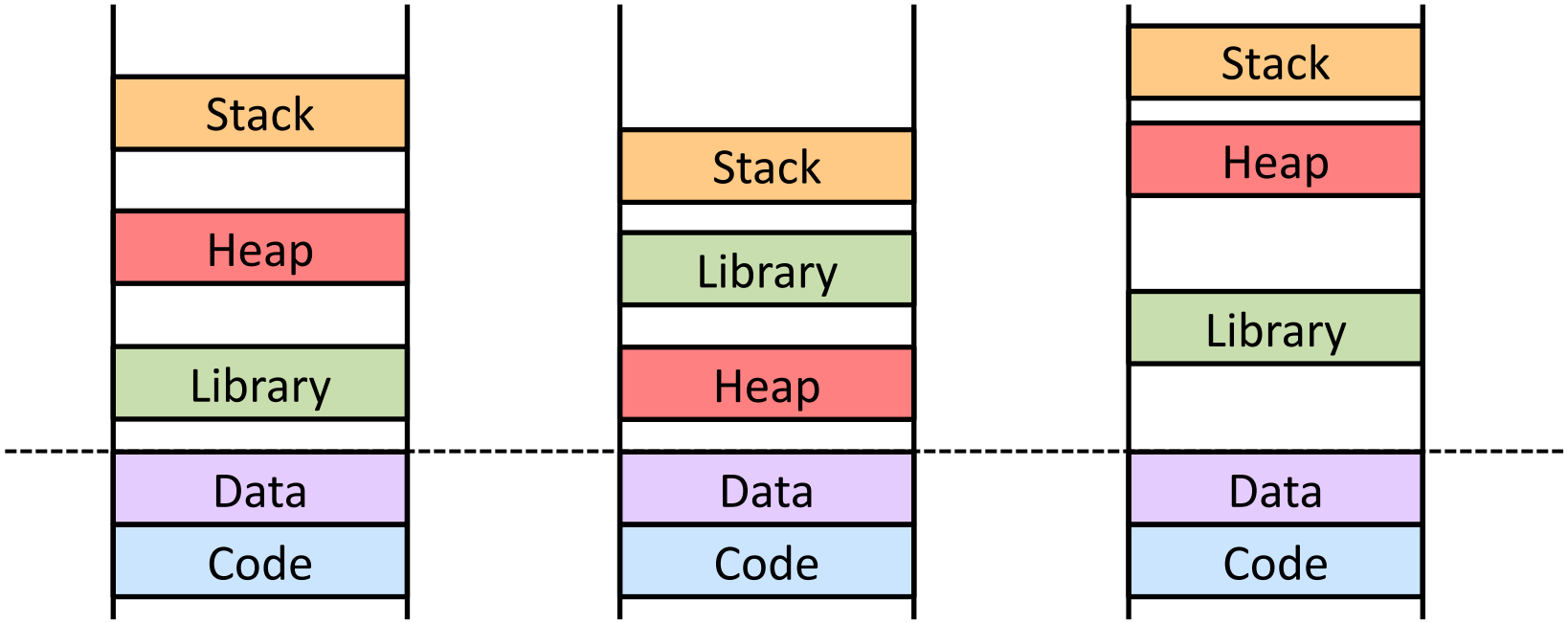
Basic Software Vulnerabilities

- **Buffer overflow (in stack and heap)**
  - Accessing out-of-bound index of an array
- **Format string bug**
  - `printf()` called with user-controllable format string
- **Use-after-free**
  - Accessing memory that has been already freed (and reallocated)
- **Race condition**
  - Time-of-check vs. time-of-use



Attacker vs. Defender

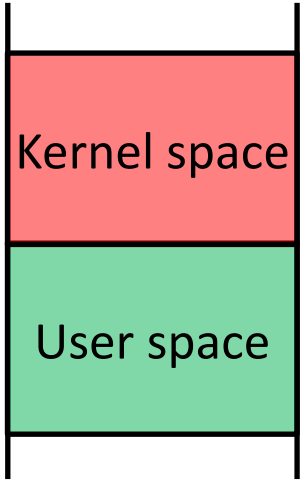
- **Mitigations: Stack canary, DEP, ASLR, …**
- **Exploit techniques: Code reuse attack, ROP, memory disclosure, …**



Kernel Security

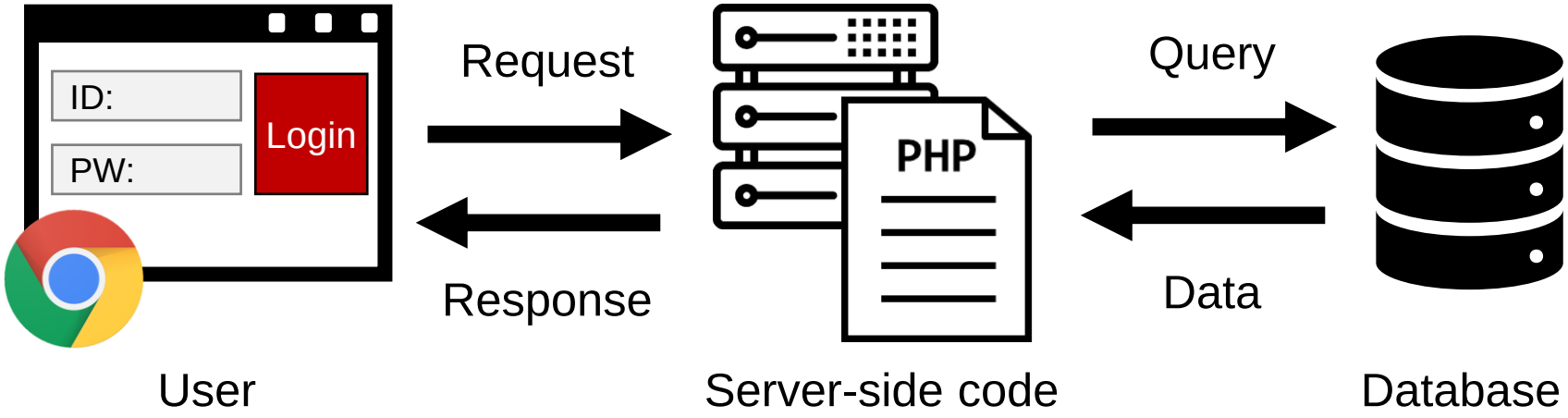
- Syscall handler should check user-provided pointers
- A specific kind of race condition called double-fetch can occur in syscall handler
- NULL dereference can be a vulnerability in kernel code (due to unique threat model in kernel exploitation)

```
// System call handler in kernel code
read_handler(int fd, void *buf, size_t n) {
  ... // Read in the file content
  memcpy(buf, file_content, n);
}
```



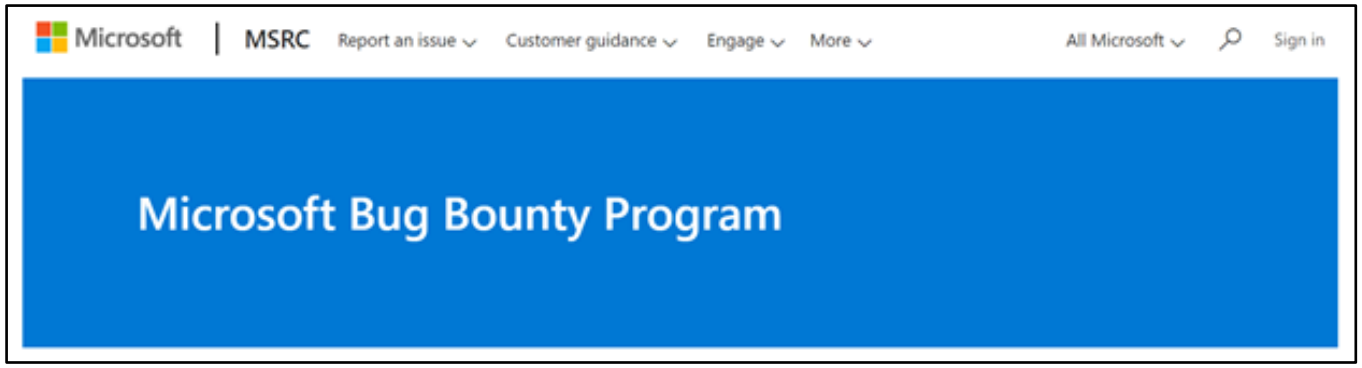
Web Security

- File upload attack
- SQL injection attack
- Cross-site scripting (XSS) attack
- Cross-site request forgery (CSRF) attack



Vulnerabilities in Real World

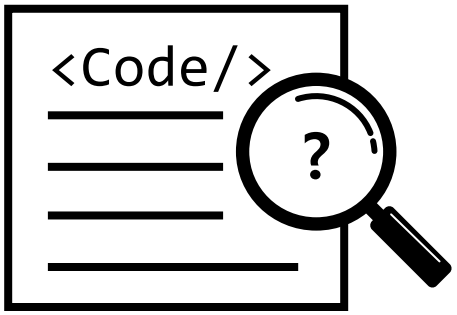
- **In this course (lab assignment), we have dealt with toy programs with artificially crafted vulnerabilities**
- **Finding bugs in real-world SW is often much harder**
  - Large and complex software, often lacking source code
- **That will be an exciting and rewarding challenge**
  - KISA's bug bounty on domestic applications
  - Google's bug bounty on Chrome browser
  - Microsoft's bug bounty on Windows kernel



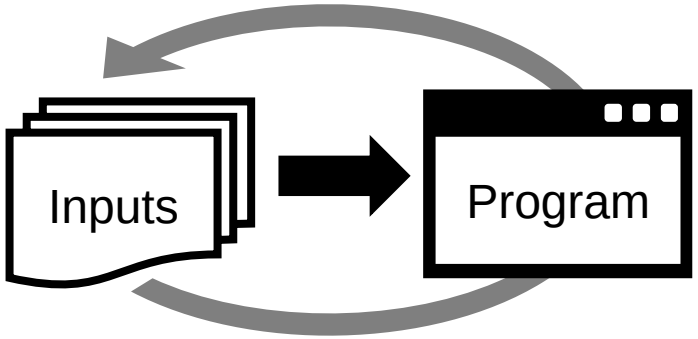
https://www.microsoft.com/en-us/msrc/bounty

Academic Research on SW Security

- **There are many interesting topics in SW security**
- **In particular, I am interested in finding software vulnerabilities automatically & systematically**
- **Our lab studies various theories and techniques in programming language and software engineering field**
  - Static program analysis, fuzz testing, symbolic execution, taint analysis … (and machine learning maybe?)



Static Analysis



Testing

Thank you