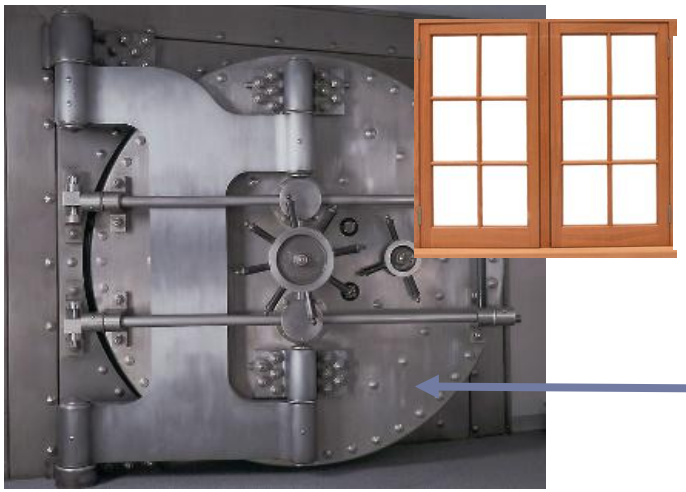
SC3010
Computer Security

Lecture 2: Software Security (I)

Basic Concepts in Software Security

Vulnerability: a *weakness* which allows an attacker to reduce a system's information assurance.

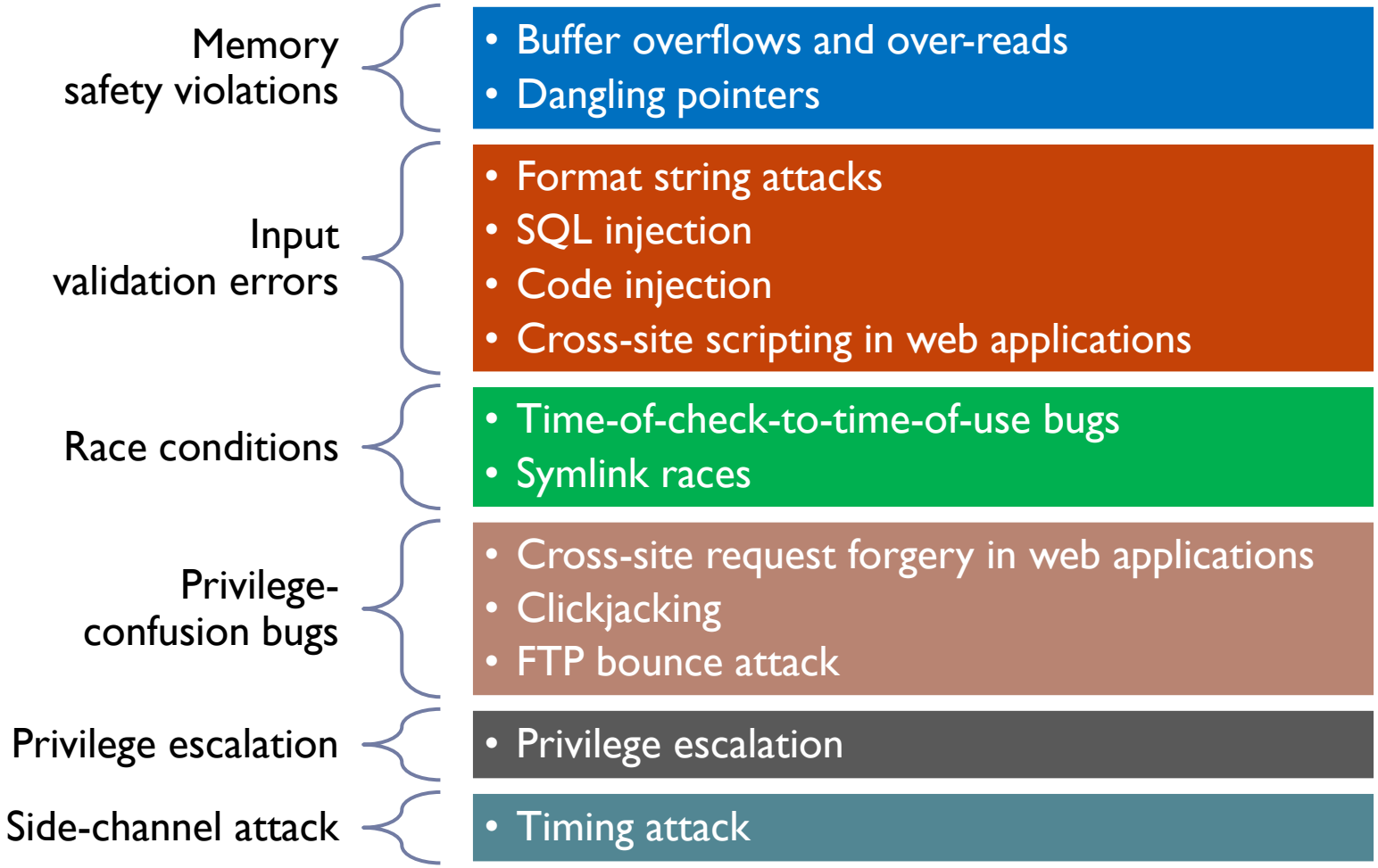


Software system

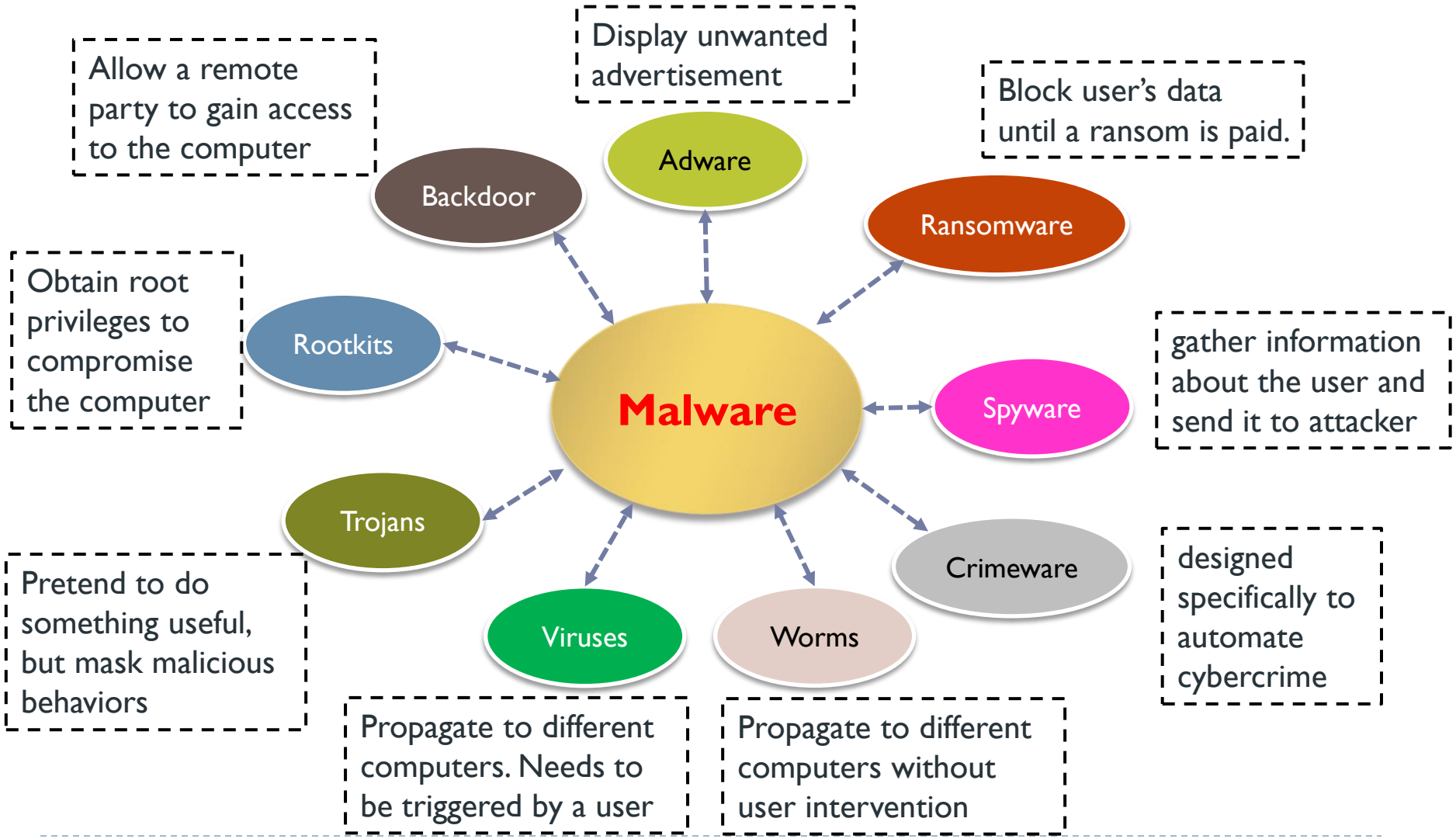
Exploit: a *technique* that takes advantage of a vulnerability, and used by the attacker to attack a system

Payload: a *custom code* that the attacker wants the system to execute

Different Kinds of Vulnerabilities



Different Kinds of Malware



Why Does Software Have Vulnerabilities

Human factor

- Programs are developed by humans. Humans make mistakes
- Programmers are not security-aware
- Misconfigurations could lead to exploit of software vulnerabilities

Language factor

- Some programming languages are not designed well for security
  - Mainly due to more flexible handling of pointers/references.
  - Lack of strong typing.
  - Manual memory management. Easier for programmers to make mistakes.

Outline

- **Review: Memory Layout and Function Call Convention**
- **Buffer Overflow Vulnerability**

Outline

- **Review: Memory Layout and Function Call Convention**
- Buffer Overflow Vulnerability

Memory Layout of a Program (x86)

Code

- The program code: fixed size and read only

Static data

- Statically allocated data, e.g., variables, constants

Stack

- Parameters and local variables of methods as they are invoked.
- Each invocation of a method creates one frame which is pushed onto the stack
- Grows to lower addresses

Heap

- Dynamically allocated data, e.g., class instances, data array
- Grows towards higher addresses