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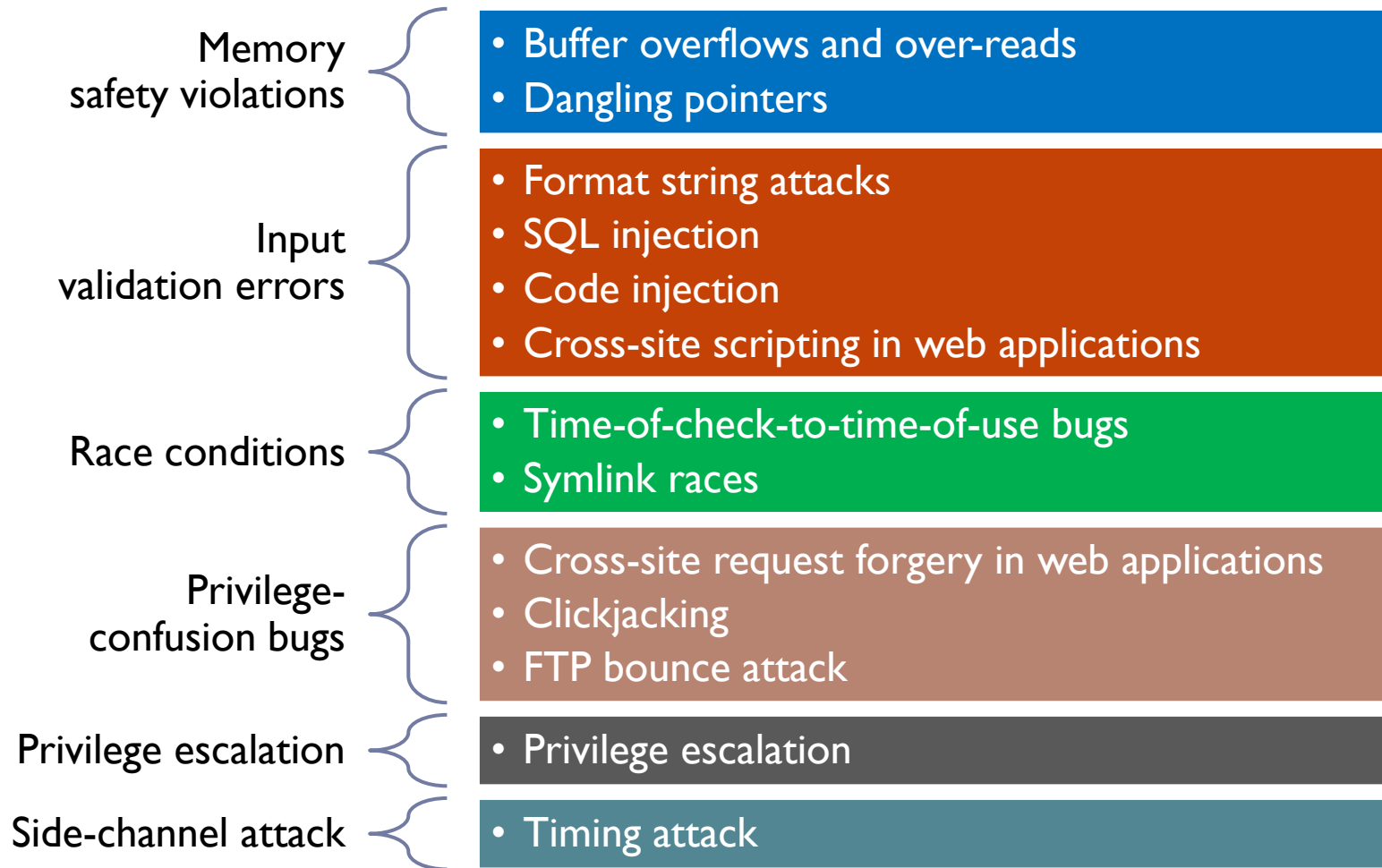
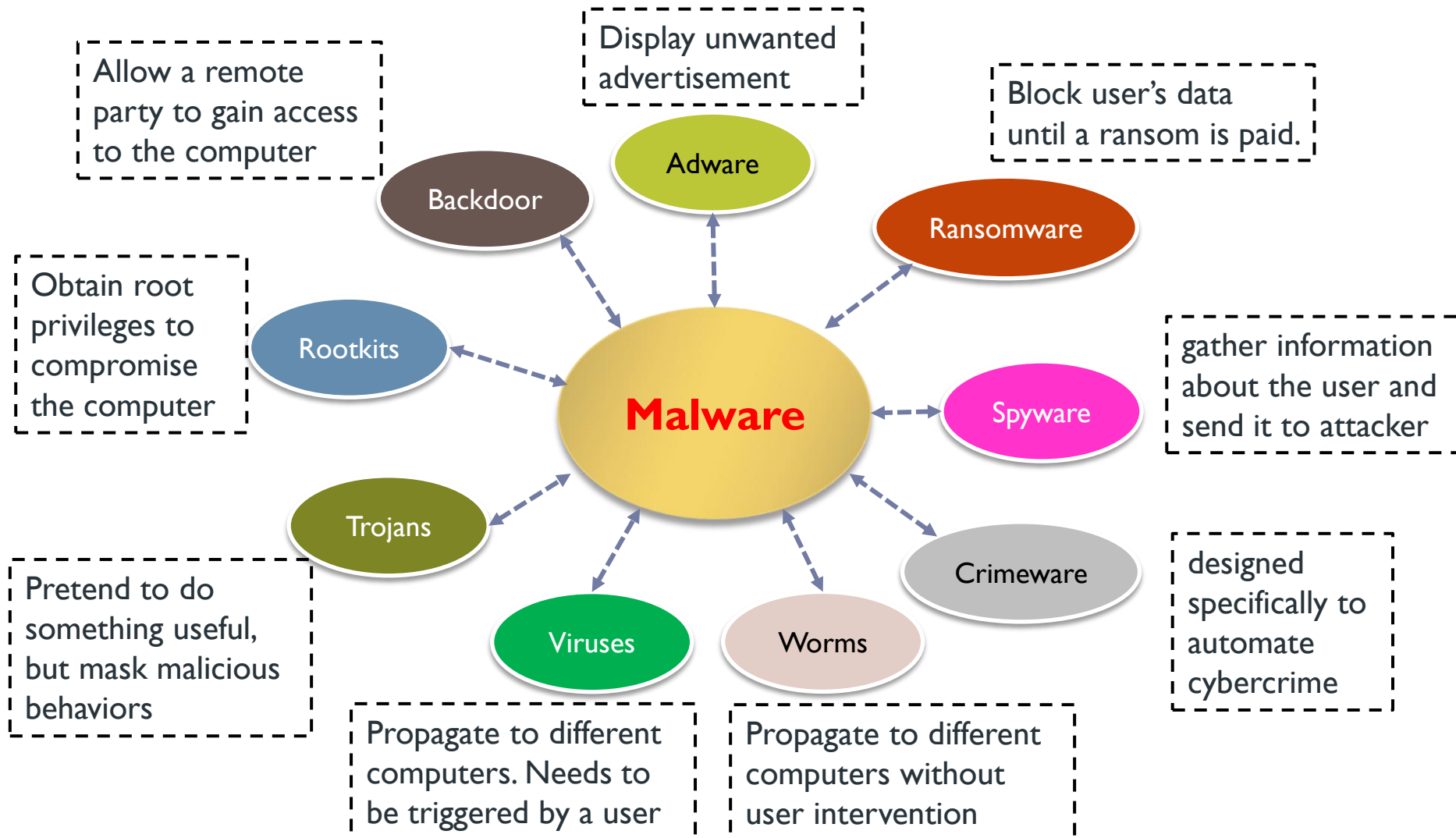**Exploit**: a _technique_ that takes advantage of a vulnerability, and used by the attacker to attack a system

**Payload**: a _custom code_ that the attacker wants the system to execute

**Software system**

# Different Kinds of Vulnerabilities

**Memory safety violations**
- Buffer overflows and over-reads
- Dangling pointers

**Input validation errors**
- Format string attacks
- SQL injection
- Code injection
- Cross-site scripting in web applications

**Race conditions**
- Time-of-check-to-time-of-use bugs
- Symlink races

**Privilege-confusion bugs**
- Cross-site request forgery in web applications
- Clickjacking
- FTP bounce attack

**Privilege escalation**
- Privilege escalation

**Side-channel attack**
- Timing attack

# Different Kinds of Malware

Allow a remote party to gain access to the computer

Display unwanted advertisement

Block user's data until a ransom is paid.

Backdoor

Adware

Ransomware

Obtain root privileges to compromise the computer

Rootkits

**Malware**

gather information about the user and send it to attacker

Spyware

Trojans

Pretend to do something useful, but mask malicious behaviors

Viruses

Worms

Crimeware

designed specifically to automate cybercrime

Propagate to different computers. Needs to be triggered by a user

Propagate to different computers without user intervention

# 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.