CS165 – Computer Security

Final Review
Dec 2, 2021

Agenda

- **Seven** Security Design Principles
  - Defense in depth
  - Least Privilege
  - Fail-Safe Defaults
  - Economy of Mechanism
  - Complete Mediation
  - Open Design
  - Separation of Privilege

- Final Review

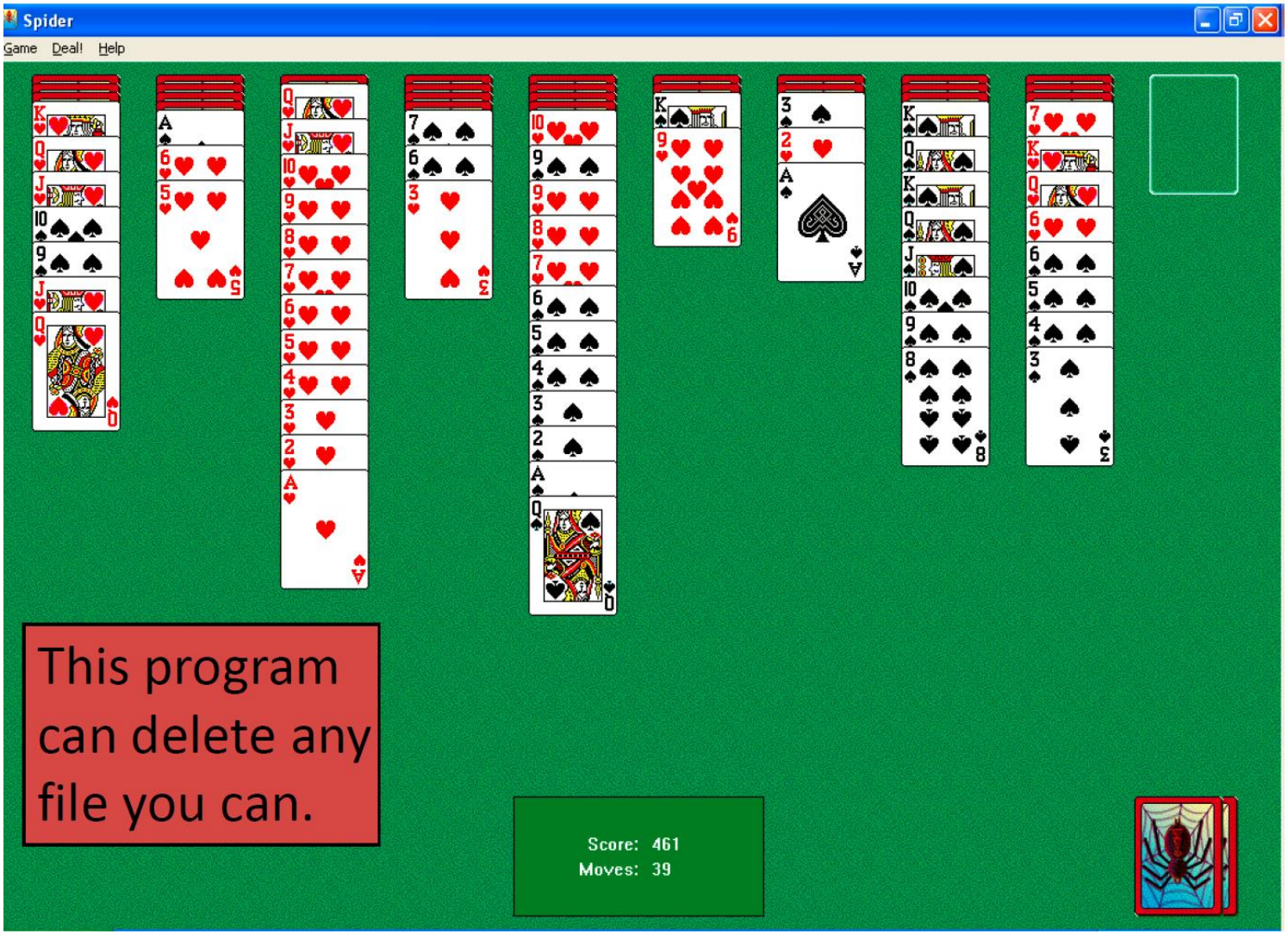
Overview

- Simplicity
  - Less to go wrong
  - Fewer possible inconsistencies
  - Easy to understand

- Restriction
  - Minimize capability
  - Minimize access
  - Inhibit communication

I. Defense in depth

II. Least Privilege



II. Least Privilege

- A subject should be given only those privileges necessary to complete its task
  - What is the task, and what is the minimal set of rights needed?
  - Rights added as needed, discarded after use

- Examples
  - "sudo" only when necessary
  - Do not open browser with root

III. Fail-Safe Defaults

- Default action is to deny access
  - Firewall

- If action fails, system as secure as when action began

IV. Economy of Mechanism

- Adi Shamir: “There are no secure systems, only degrees of insecurity.”
- “No system is completely, 100% secure against all attacks. Rather, systems may only need to resist a certain level of attack. There is no point buying a $10,000 firewall to protect $1,000 worth of trade secrets.”




IV. Economy of Mechanism

- Keep it as simple as possible
  - KISS Principle
- Simpler means less can go wrong
  - And when errors occur, they are easier to understand and fix
- Interfaces and interactions
- Example
  - Stateful TCP firewall introduces the vulnerability because it tries to be fancy

V. Complete Mediation

V. Complete Mediation

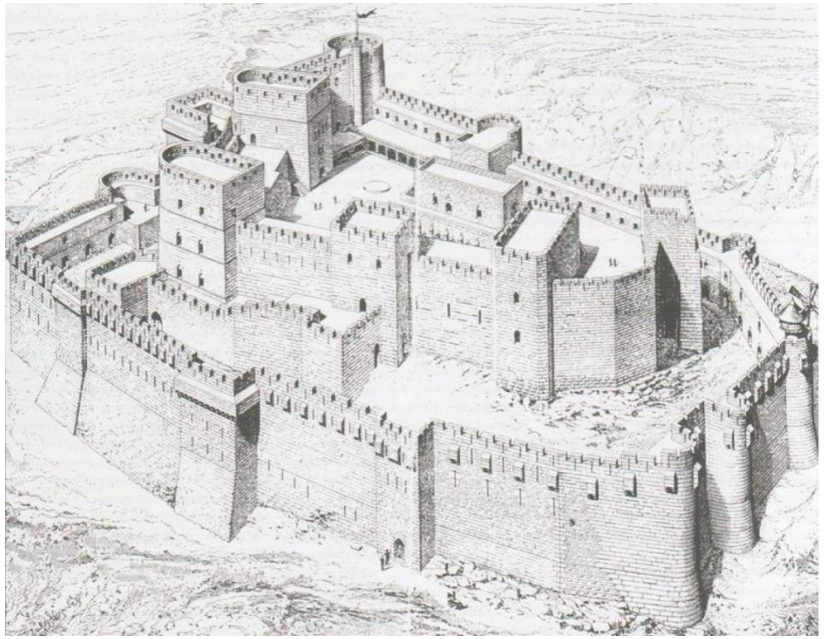
- Check every access
- Usually done once, on first action
  - UNIX: access checked on open, not checked thereafter
- If permissions change after, may get unauthorized access

VI. Open Design

- Security should not depend on secrecy of design or implementation
  - Secrecy != Security
  - Complexity != Security
  - “Security through obscurity”
  - Caveat: does not apply to “data” such as passwords or cryptographic keys

VII. Separation of Privilege

- Require multiple conditions to grant privilege
  - Separation of duty
    - "Company checks over $75,000 need to be signed by two officers."
  - Defense in depth



Summary

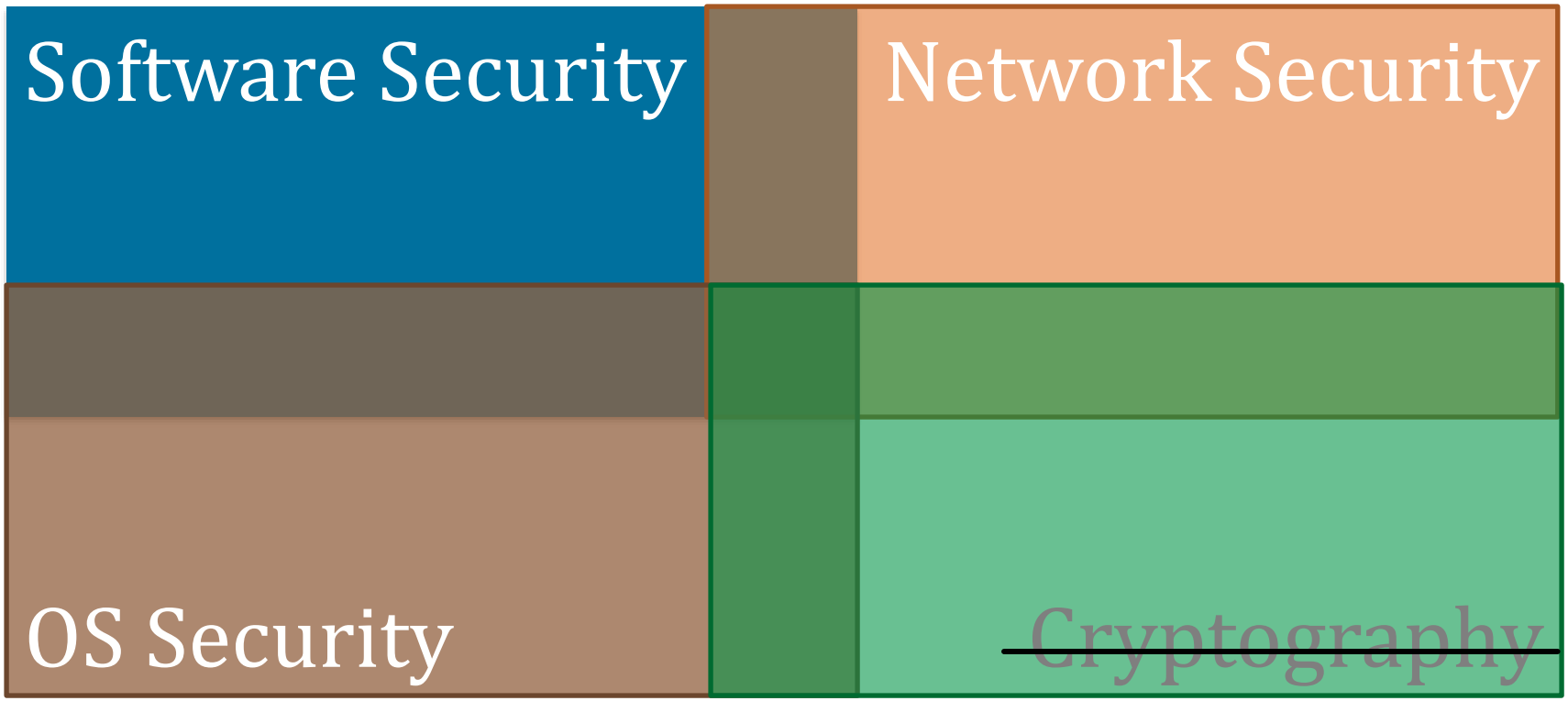
- Principles of secure design underlie all security-related mechanisms
- Require:
  - Good understanding of goal of mechanism and environment in which it is to be used
  - Careful analysis and design
  - Careful implementation

Agenda

- Seven Security Design Principles
  - Defense in depth
  - Least Privilege
  - Fail-Safe Defaults
  - Economy of Mechanism
  - Complete Mediation
  - Open Design
  - Separation of Privilege

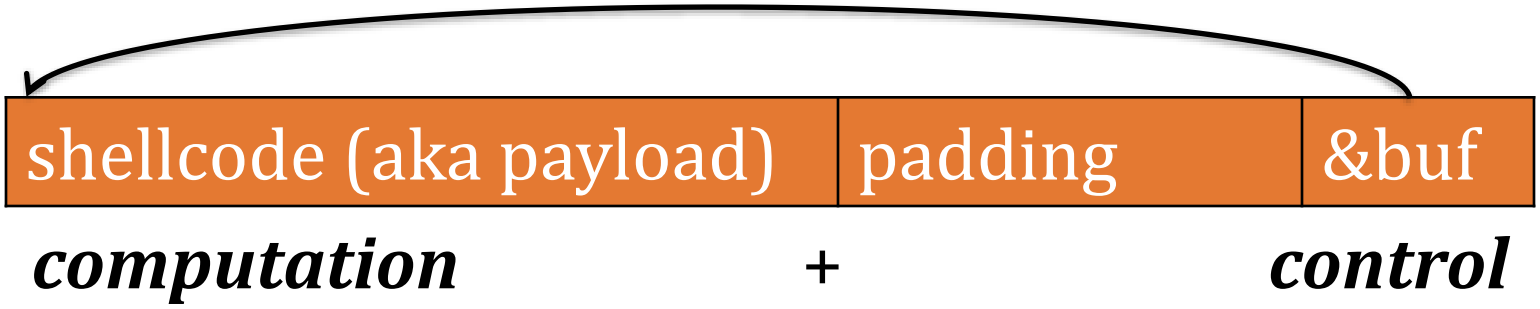
- Final Review

This Class: Introduction to the Three Cornerstones of Security



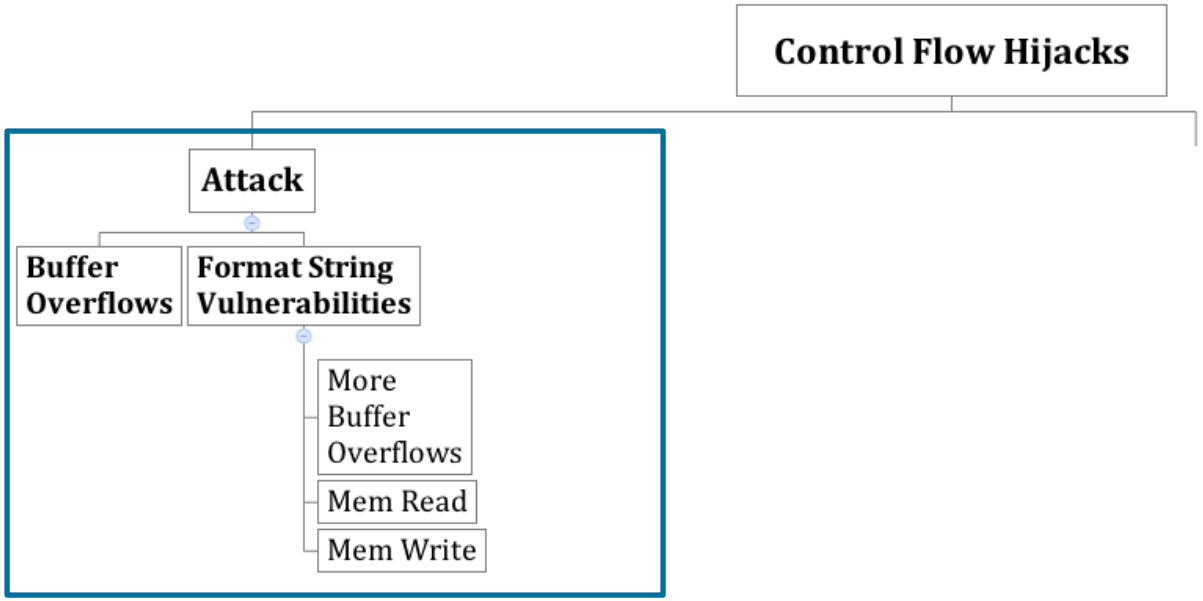
Software Security

Control Flow Hijacks



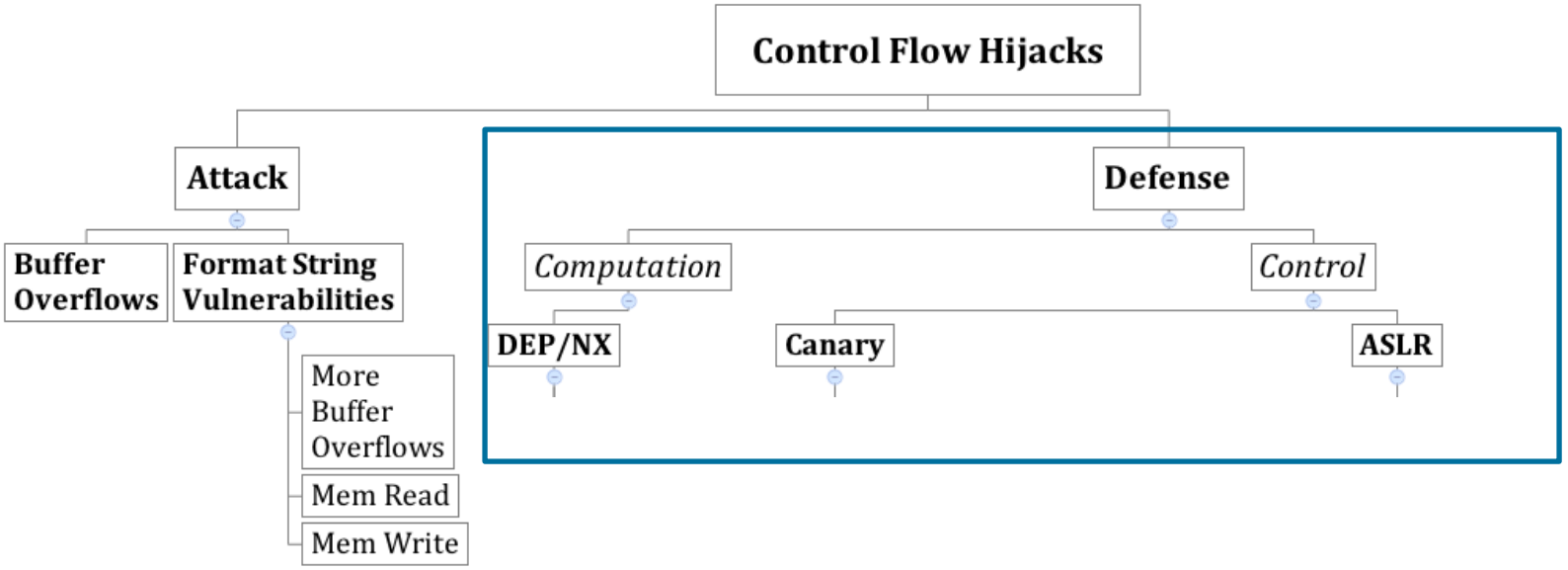
Allow attacker ability to run arbitrary code

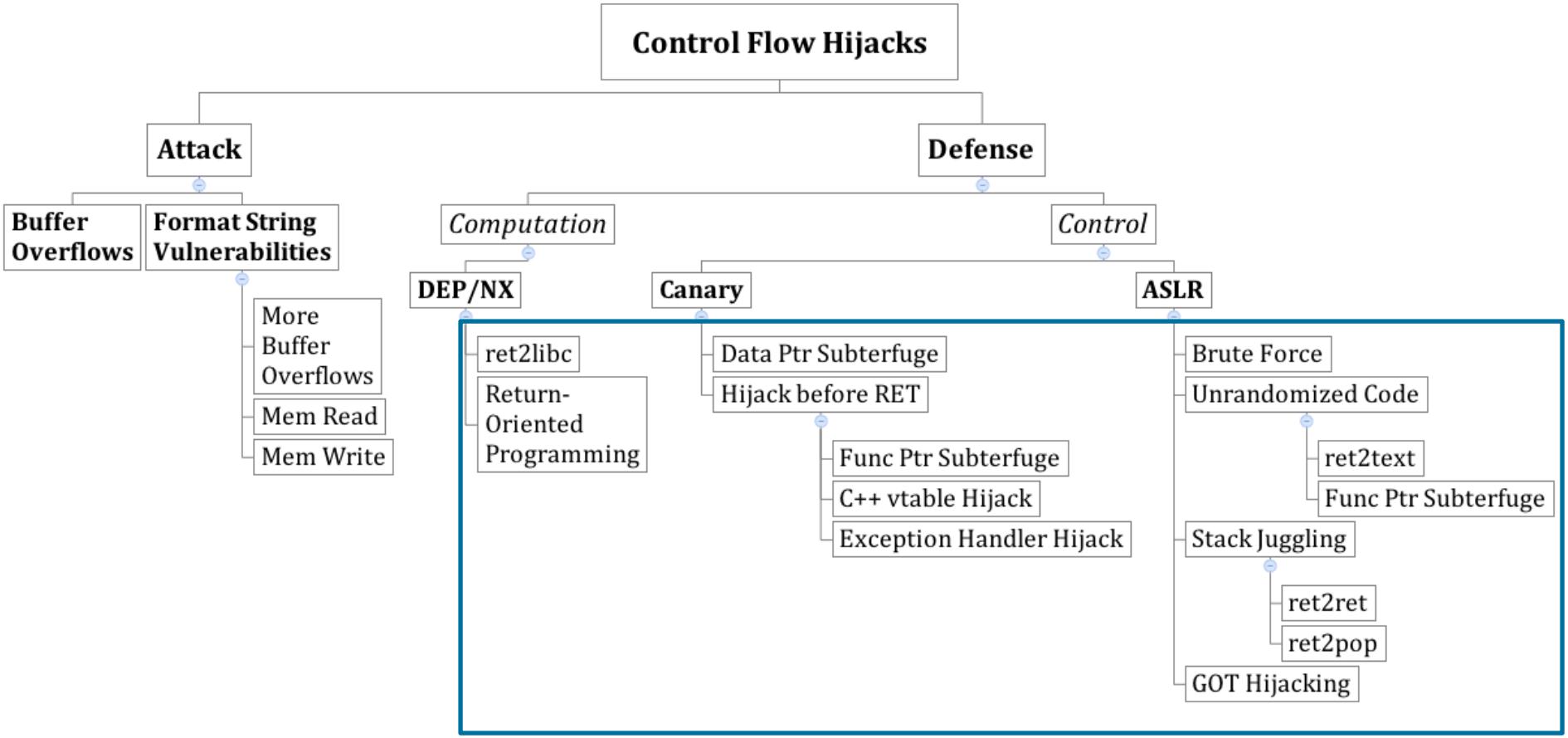
- Install malware
- Steal secrets
- Send spam
- …


Control Flow Hijacks
Attack
Buffer Overflows
Format String Vulnerabilities
More Buffer Overflows
Mem Read
Mem Write

Attacks

- Stack buffer overflow
  - Shell code injection, ret2libc, ROP, blind ROP
- Heap buffer overflow
  - Memory write
- Integer overflow
  - Can be turned into buffer overflow
- Format string vulnerability
  - Memory read/write


Control Flow Hijacks
Attack
Defense
Buffer Overflows
Format String Vulnerabilities
More Buffer Overflows
Mem Read
Mem Write
Computation
Control
DEP/NX
Canary
ASLR


Control Flow Hijacks
Attack
Defense
Buffer Overflows
Format String Vulnerabilities
More Buffer Overflows
Mem Read
Mem Write
Computation
Control
DEP/NX
Canary
ASLR
ret2libc
Return-Oriented Programming
Data Ptr Subterfuge
Hijack before RET
Func Ptr Subterfuge
C++ vtable Hijack
Exception Handler Hijack
Brute Force
Unrandomized Code
ret2text
Func Ptr Subterfuge
Stack Juggling
ret2ret
ret2pop
GOT Hijacking

Software Security

- Recognize and exploit vulnerabilities
  - Buffer overflow
  - Format string
  - Gist of other control flow hijacks, e.g., integer overflow, heap overflow
- Understand defenses in theory and practice
  - ASLR
  - DEP
  - Canaries
  - Know the limitations!

Software Security

- Attack surface definition
  - Adversary-controlled entry points
  - System-level vs. Program-level attack surface
  - For each program, library calls / syscalls can be viewed as potential attack surface
- Threat model definition
  - Assumption about attackers' resources/capabilities/goals
- Ability to reason about attack surface given a **system** and **threat model**
  - Needed for both attacker and defender
  - Questions will be asked

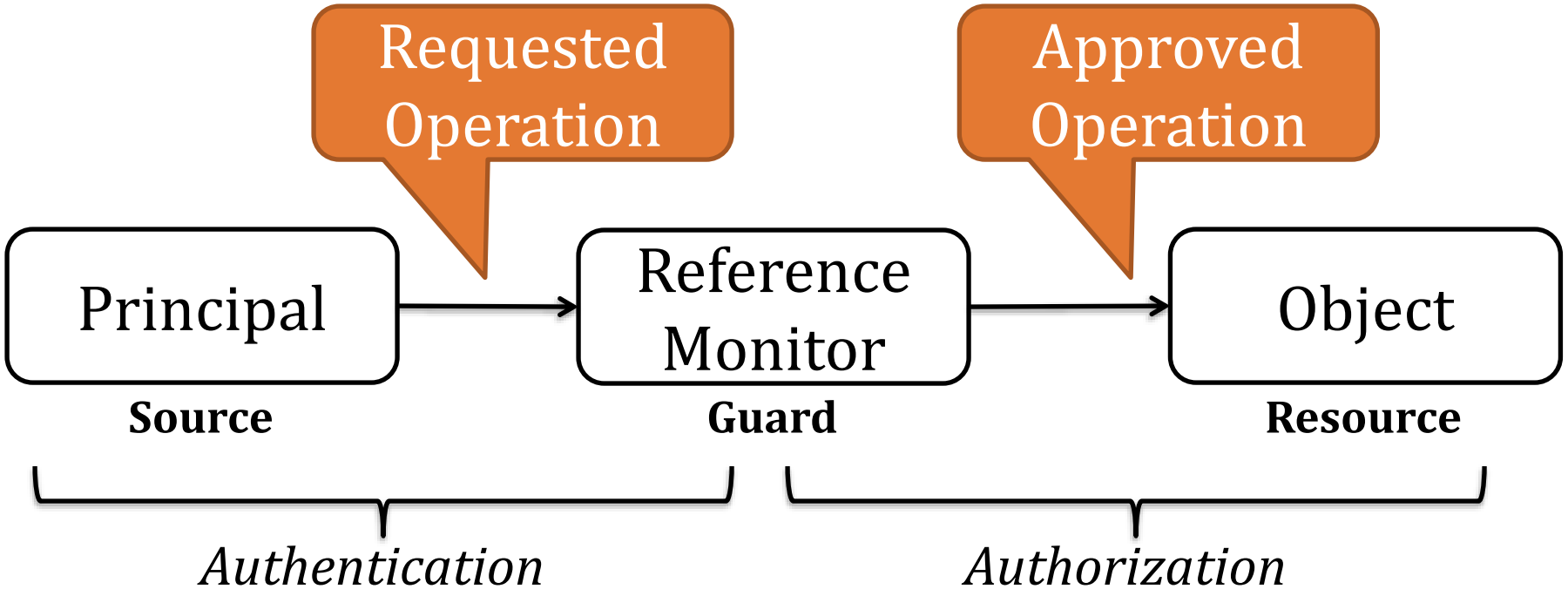
Software Security

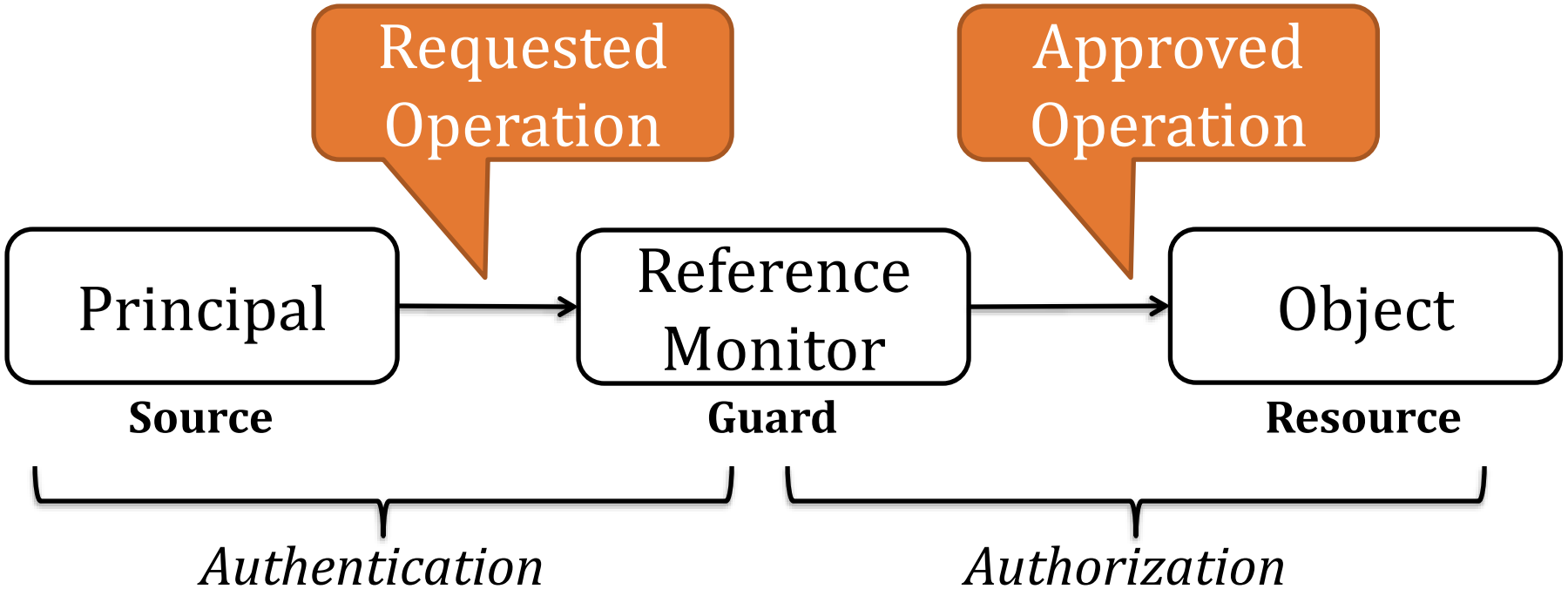
- Program analysis techniques
  - Dynamic vs. Static (understand pros and cons)
  - Fuzzing (multiple types)
  - Static analysis basics: abstract program executions (given description of an analysis technique, understand what it can achieve, e.g., what bugs can be found)
  - Information/Data flow analysis

Software Security

- Control-Flow Integrity (CFI) and Software Fault Isolation (SFI)
  - Basic principle of CFI
    - Restrict indirect control transfer targets
    - Why it works and cannot be subverted
  - Basic concept of SFI
    - Memory isolation within a process

OS Security


Requested Operation
Approved Operation
Principal
Reference Monitor
Object
Source
Guard
Resource
Authentication
Authorization

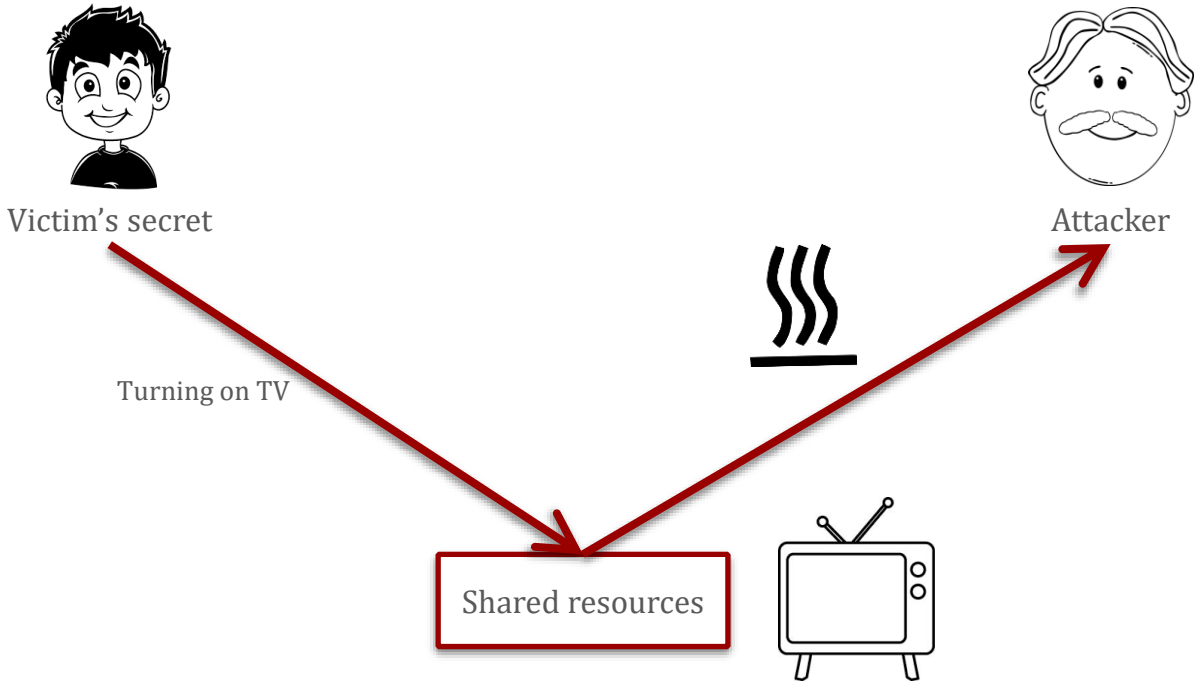


In security, we isolate reasoning about the guard

OS Security

- Authentication & Authorization
  - Principles
  - Reference monitors
  - Access control lists
- Information flow security
  - High secrecy object ---x---> Low secrecy subject
  - Low integrity object ---x---> High integrity subject
  - Program analysis can do fine-grained checking
- Resource access vulnerabilities
  - Mismatch in expectation of the secrecy or integrity of objects

Side Channels



Side Channels

- Fundamental reason
  - Shared resources between a victim and attacker
  - Victim's secret propagation (information flow) to shared resource, and then to attacker
- Examples
  - Shared global variable among sockets
    - Global rate limit

Network Security

Network Security

- Threat models
  - Passive, MITM, Off-path
- IP Spoofing
  - Lack of accountability at the lowest-level
- DNS poisoning attacks
- TCP sequence number inference attacks
- Firewall – filtering rules
- Intrusion Detection
- Denial of service

Other topics

- Vulnerability research
- IoT
- Forensics
- Underground economy