CS5231: Systems Security

Lecture 1: Overview

About This Course

- (Generalized) definition of systems
- Principle and practice of systems security
  - Understanding security principles through practice
  - Learning skills of programming, system administration, and etc.
- Research frontier of systems security

Uniqueness of This Module

- Think in a different angle
  - How various systems can fail?
  - How to prevent such failures?
- Learn to think like a hacker, behave like a defender
  - Make no assumptions of hackers
- Heavily based on system programming
  - Have fun!

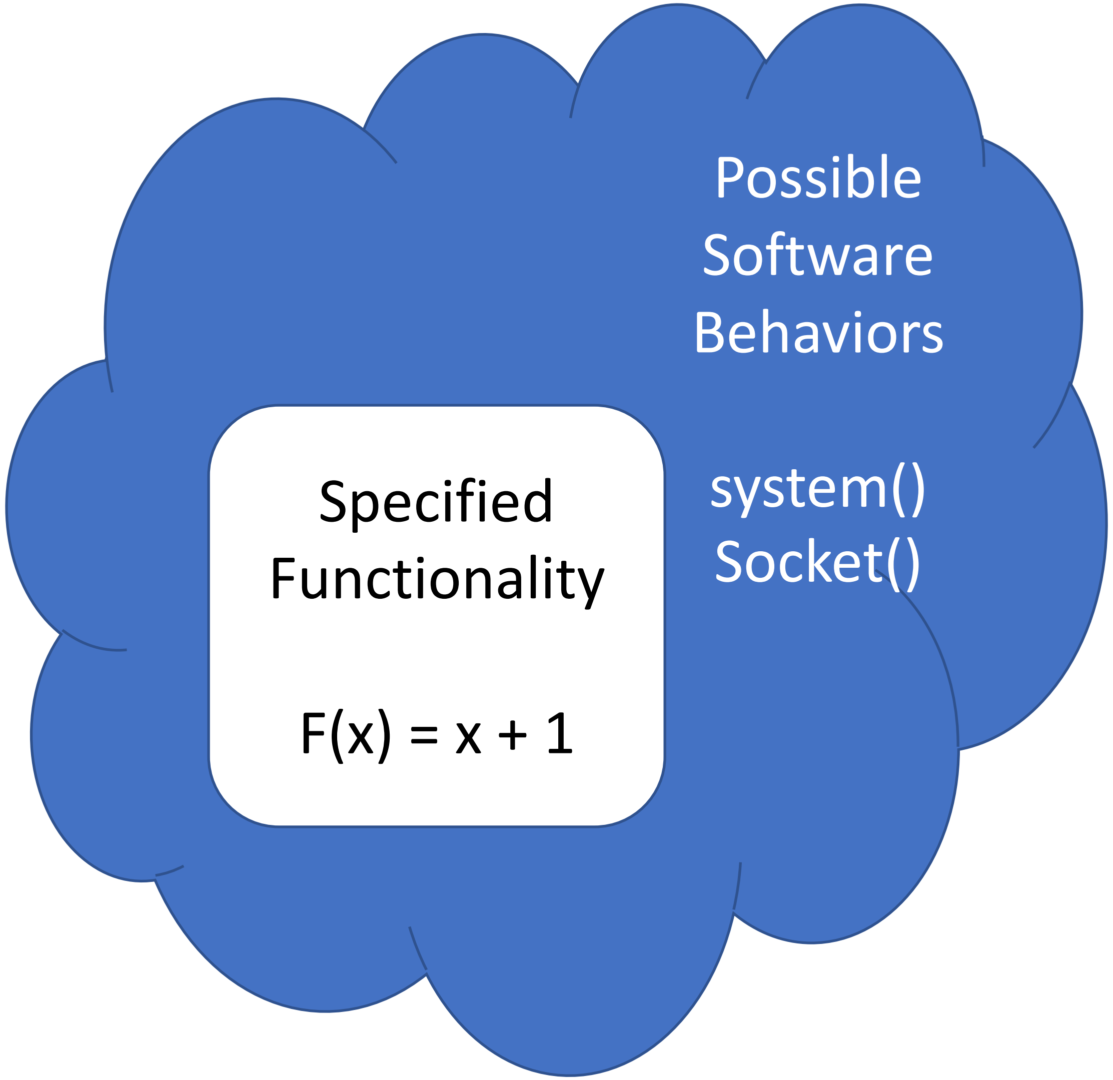
The Security Problem

What are the recent security incidents in news?

Why Does This Happen?

- Functionality: the primary concern during design and implementation.
  - Security is the secondary goal
  - Unawareness of security problems
- Unavoidable human mistakes
  - Awareness
  - Lazy programmer
- Complex modern computing systems

Security: Mission impossible



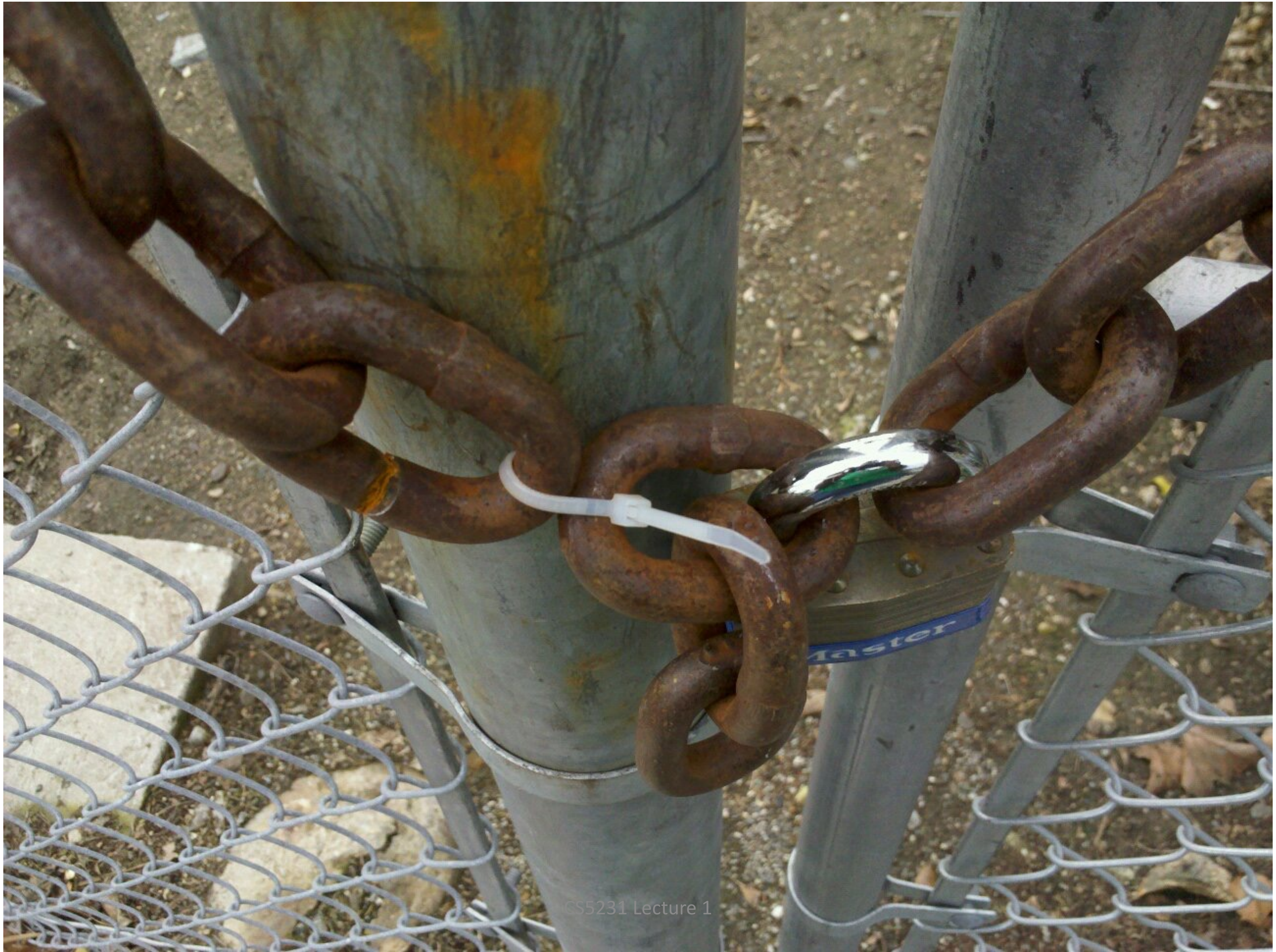
- But in practice, we need to make the security problem under control.
- Need better understanding of whole system

The Axioms of Security

Principle of Easiest Penetration

- Security is about every aspect of a computing system
  - Hardware, software, data, and people.
- Principle of easiest penetration:
  - Any system is most vulnerable at its *weakest point*.
  - Attackers don't follow any rules. Don't underestimate their creativity.

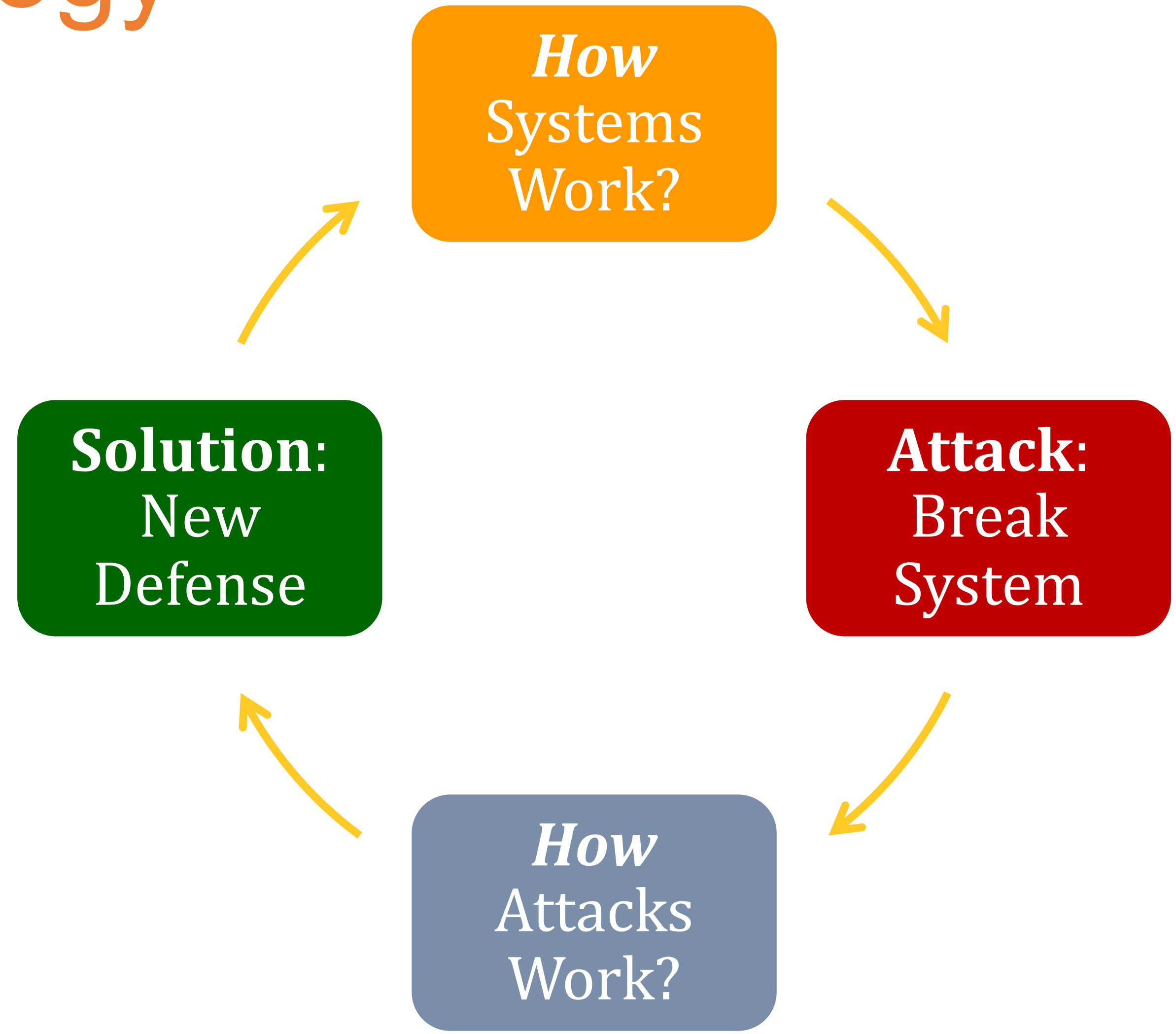
Security can be no stronger than its weakest link.



Master

Methodology of Security

Methodology



How Systems Work?

Attack: Break System

How Attacks Work?

Solution: New Defense

Learning to Attack

- If you know the enemy and know yourself, you need not fear the result of a hundred battles.

  知己知彼，百战不殆。

  Sun Tzu, Art of War

- To prevent attack, we need to learn how attack happens

Ethical Use of Security Information

- We discuss vulnerabilities and attacks
  - Most vulnerabilities have been fixed
  - Some attacks may still cause harm
  - Do *not* try these at home
- Purpose of this class
  - Learn to prevent malicious attacks
  - Use knowledge for good purposes

Administrative Issue

Administrative Issues

- In-class tests/quiz: 30%
- Individual assignments: 45%
  - Three homework assignments
- Final group project: 25%
- No final exam

Individual Homework Projects

- Sample topics of programming assignments
  - Memory error and attacks
    - Assembly, C, gdb
  - System auditing and provenanace
  - Linux kernel security mechanisms
    - Linux kernel programming
    - Linux security modules, eBPF

Group-based Final Project

- Project Goal:
  - Apply our methodology: Deeply understand of a large system, understand attacks, and design solutions.
- A typical group has 2-3 students.
  - Find teammates with similar interest, e.g., binary, kernel, etc.
  - Based on the same base system, develop solutions with individual components to ***understand or solve*** security problems
  - Please announce your group information to the TA mailing list
  - If you need to form group of three students, a concrete proposal with individual contributions is needed.

Notifications & Communication

- Watch out for Canvas announcements
- You are expected to participate in in-person lectures.
    - Interactions beyond lecture notes…
- Please use email cs5231ta@googlegroups.com with for all email communication related to the module.
- Teams Channel “Consultation” for general consultation, private message for quick-response matters

Honesty & Collaboration

- TA and instructor will not "see / debug" code
- All questions go to Canvas forum and Teams Consultation

- Academic Honesty
  - You may discuss high-level approach to solving or share public sources of information via the forum.
  - But, independently solve the assignment
  - Not OK to find answers to the assignment questions (past students, instructors, other students, friends, Internet)

- Ethics: Responsible Disclosure
  - If you find a system vulnerable, inform the company / team responsibly
  - Not ok to exploit or sell vulnerability information.

Academic Dishonesty

A simple rule in NUS:

If reported or caught cheating, in any way, all students involved will get an **F grade**

- Plagiarism is a serious offense in academia
- Information for plagiarism definition and prevention
  - `http://www.cit.nus.edu.sg/plagiarism-prevention/`
- We use the *Turn It In* tool to check all submissions
  - Submissions are compared with document on the Internet and against one another

Prerequisites

- Have basic knowledge of:
  - OS, Architecture, Compilers, Systems Programming, Basics of Probability Theory
- Have worked at some point with:
  - C/C++ programming
  - Tools like Linux commands, GDB (see notes)

- Many who take this class don't have the full coverage of these pre-requisites. That is fine. Prepare to pick up the requisite knowledge as you need them.

One more thing

The view angle of system security researchers

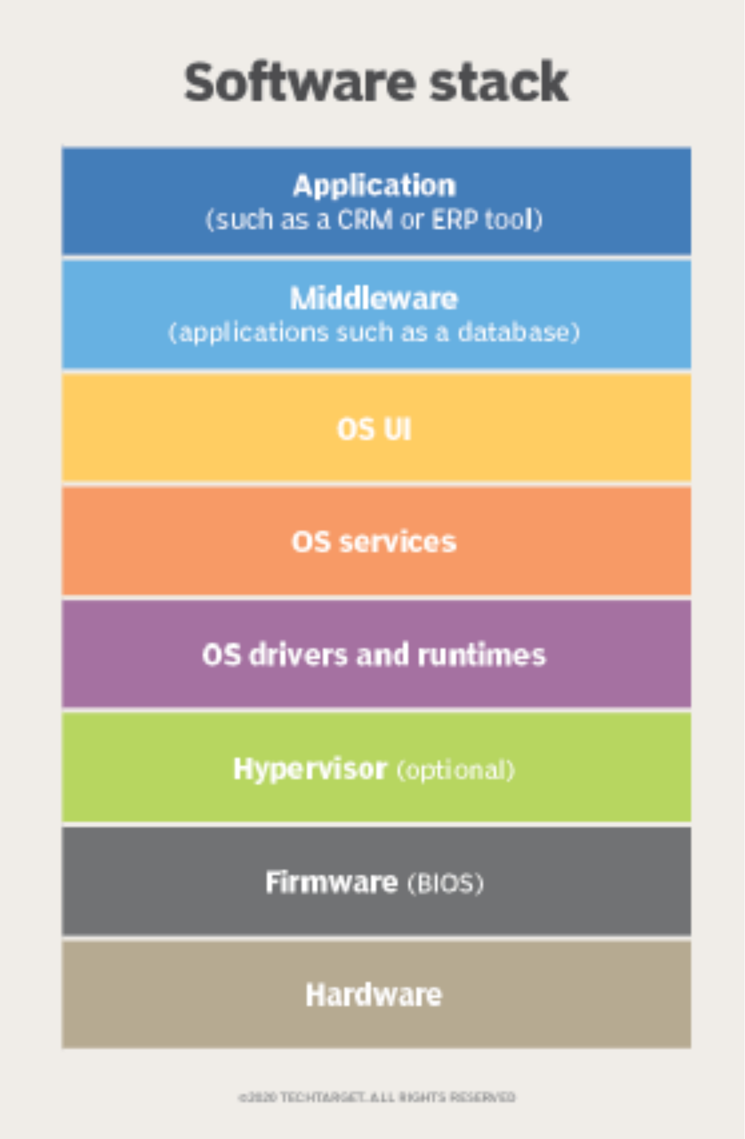
The view of systems

Program/App

System Services

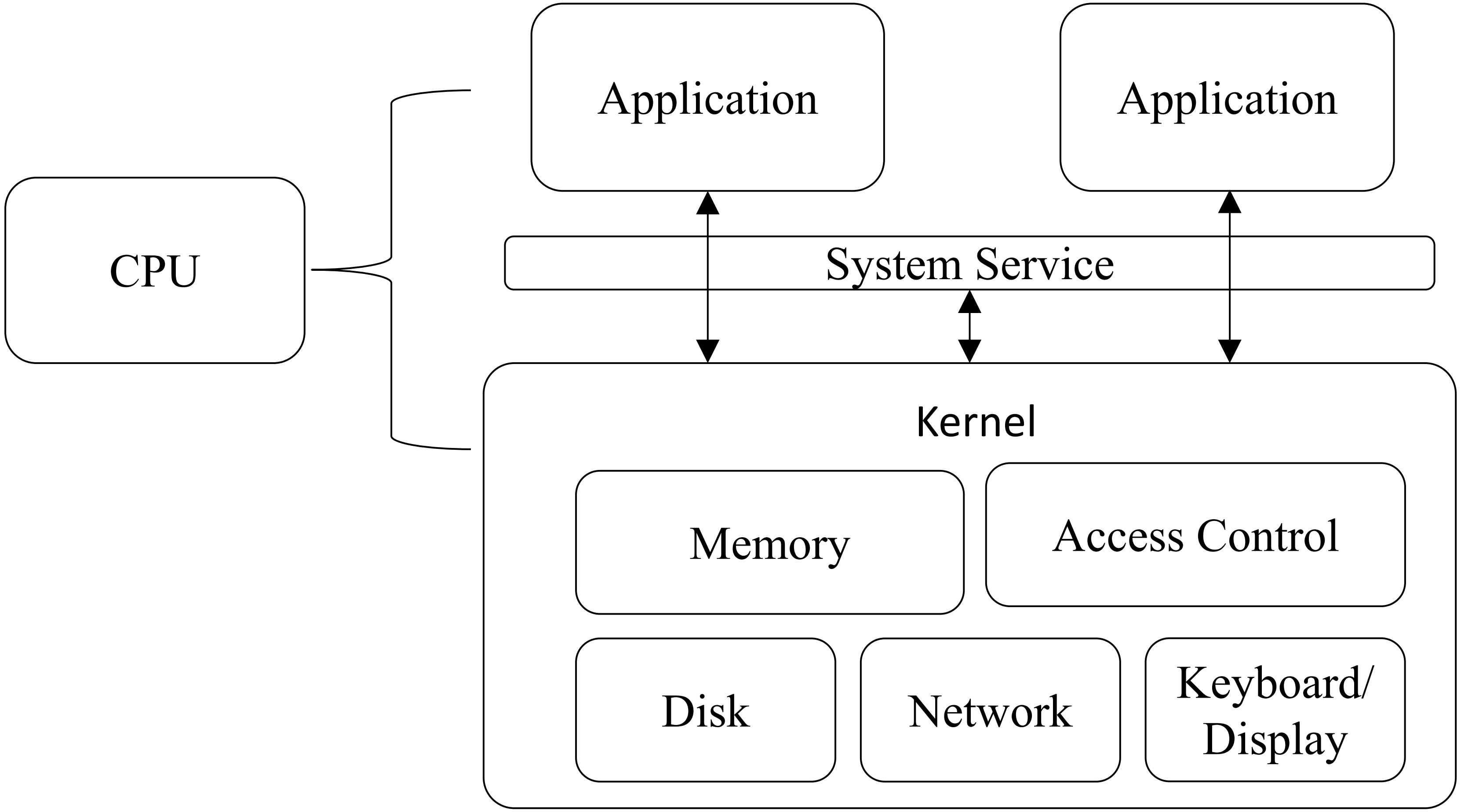
System Kernel

Hardware/Execution Engine

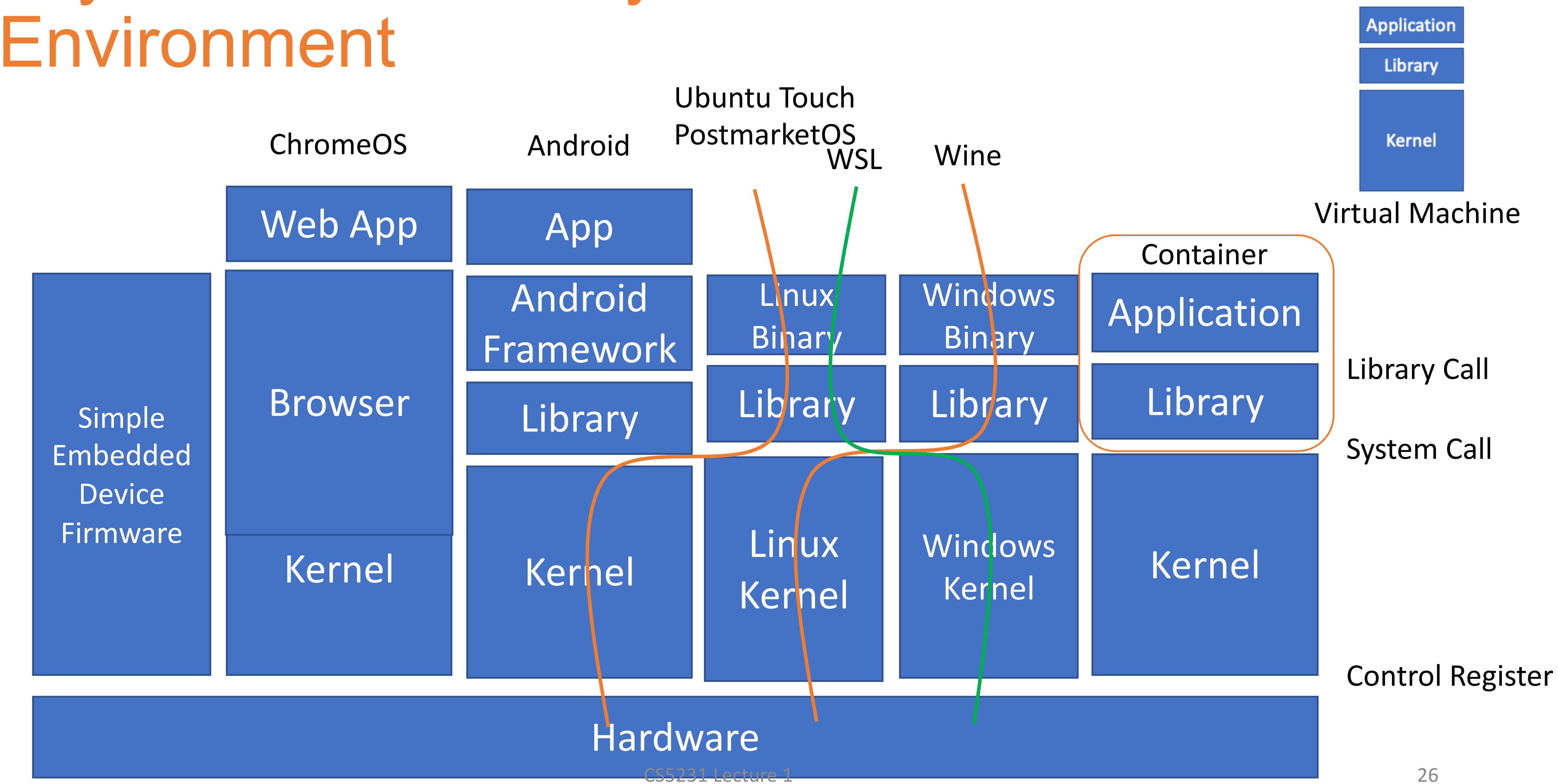


©2020 TECHTARGET. ALL RIGHTS RESERVED

Component View of Operating System



Layers and Flexibility of Execution Environment



The Key Question about Security

Authentication
- Who is who?

Response

Who can do what? → **If not, what to do?**

Security policies

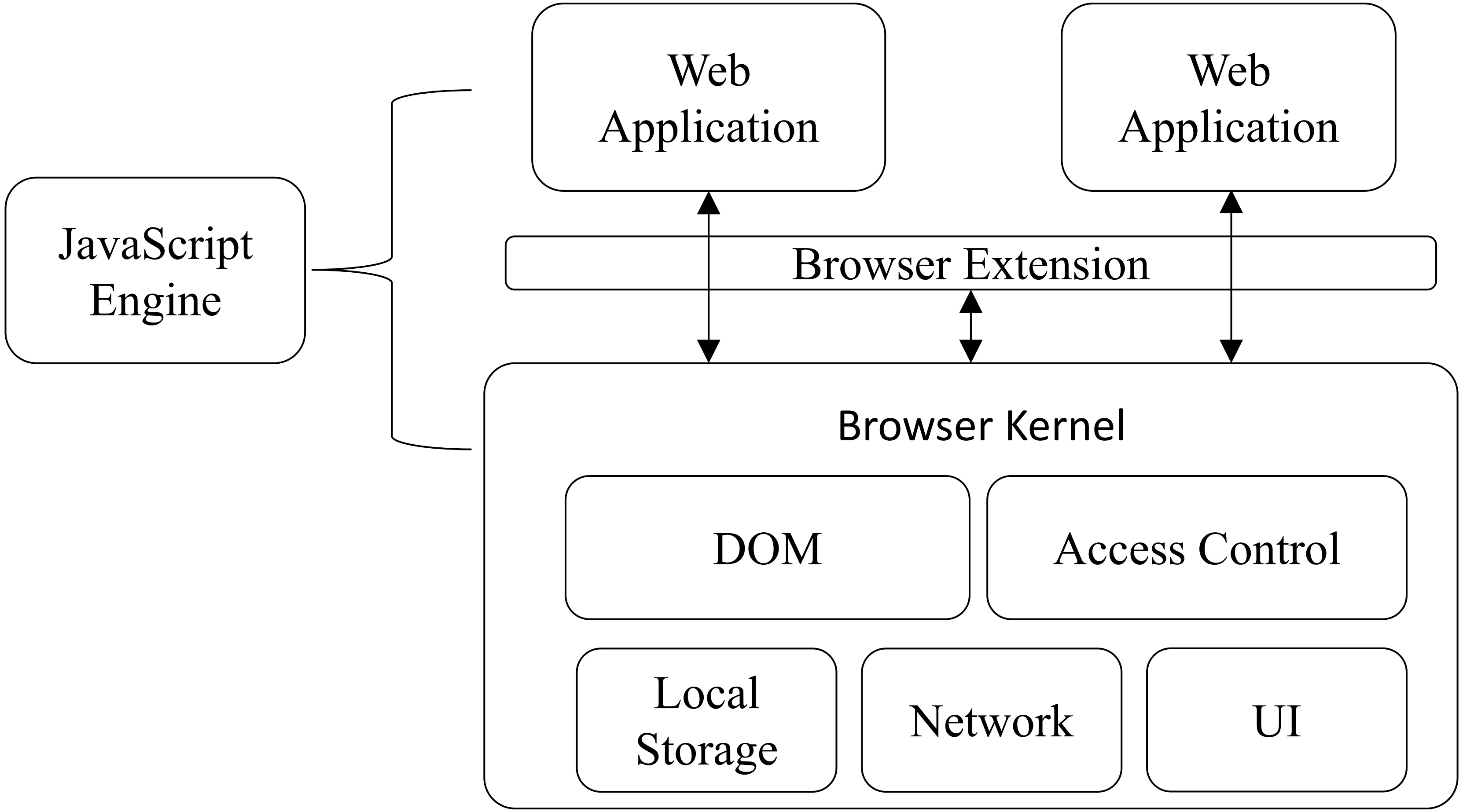
Runtime/Execution Environment

Prevention

Our Lens from System Angle: Guiding Questions

- What is the runtime platform/code/data?
- What are the owners and how to identify them?
- What are the resources to be protected?
- What is considered a security problem?
- What is the nature of the protection mechanism?
  - Access control
  - Isolation
  - Deterrence
  - …

Component View of Browser

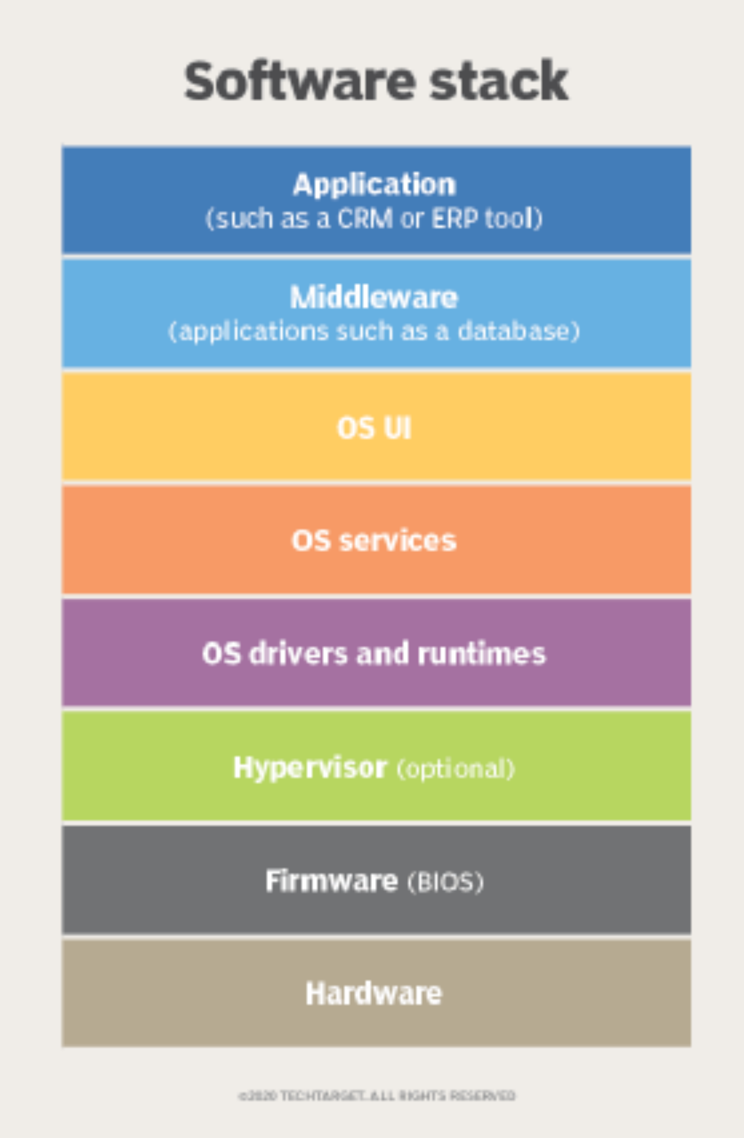


Emerging Systems

- LLM
- Web3
- Industry control systems
- …

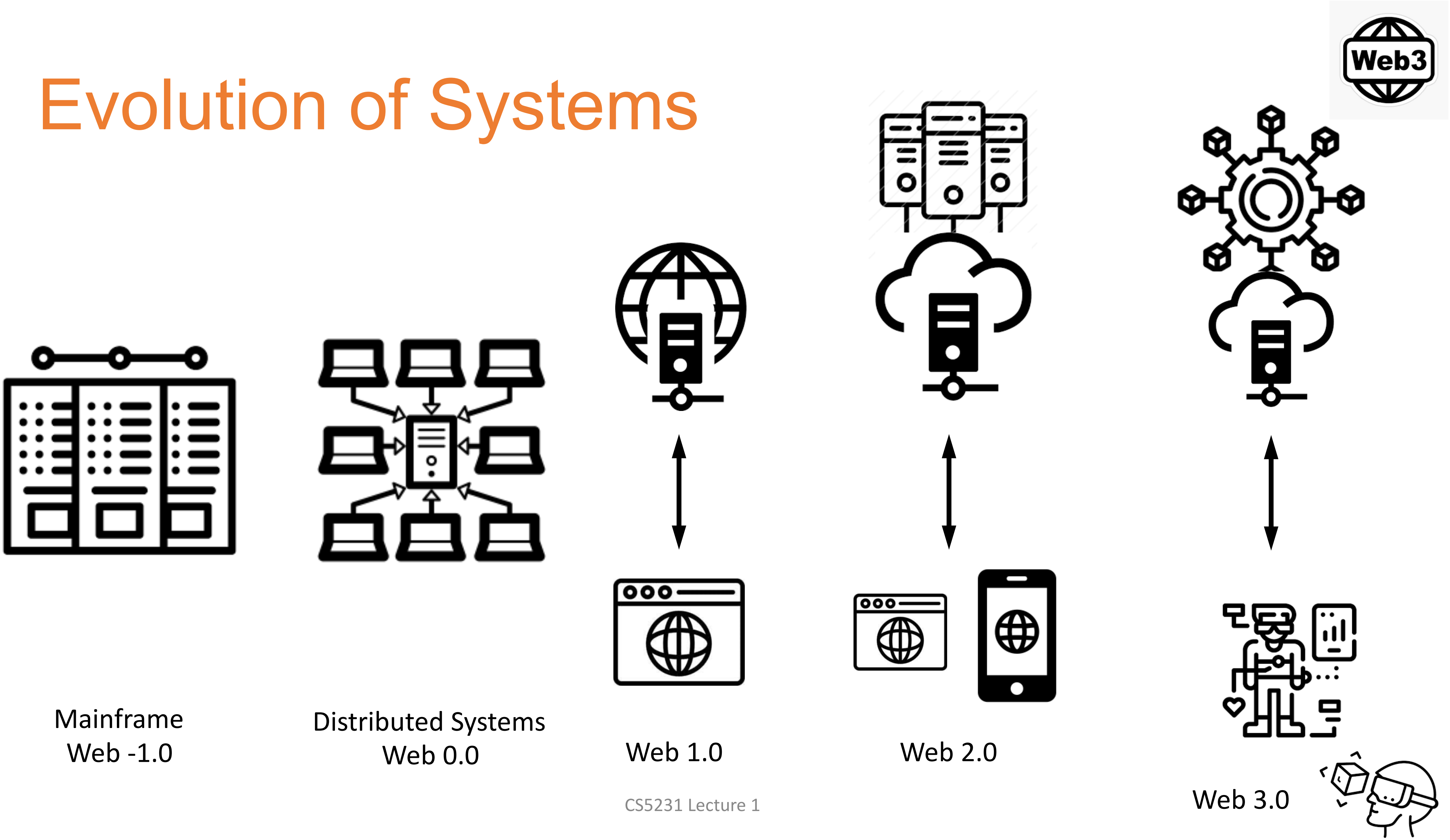
LLM Security

- LLM as AI process
  - Loss, adversarial sample, distribution, …
- LLM as software
  - Components and functions, call paths
- LLM as system
  - CPU, memory, user isolation



©2020 TECHTARGET. ALL RIGHTS RESERVED

Evolution of Systems



Mainframe
Web -1.0

Distributed Systems
Web 0.0

Web 1.0

Web 2.0

Web 3.0

Open discussion of ideas and topics

Thanks!
See you next week…