Information Assurance and Security – ACIT 4630

Hesam Alizadeh

Week 2 – Winter 2024

LAB report and demo

- Practice presentation skills
- Expectations:
  - Both be present
  - Be ready to explain what you have done
  - Show proper understanding
- Submit one report per group

Learning Outcomes

- List different sources of vulnerabilities and unique challenges associated with each
- Describe the importance of including security in software architecture design
- Explain the process of vulnerability management in an organization

Vulnerability Sources

- Server vulnerabilities
- Endpoint vulnerabilities
- Supply chain vulnerabilities
- Configuration vulnerabilities
- Architectural vulnerabilities

Server vulnerabilities

Weaknesses and flaws in web servers or other server-side systems.

- Outdated software / unsupported OS
- Buffer Overflow Vulnerabilities
- Privilege Escalation Threats
- Insecure Protocols (e.g., FTP, Telnet, POP3)
- Servers configured for debugging modes

Endpoint vulnerabilities

Weaknesses associated with individual devices connected to a network.

- Like servers but with stricter configuration
- Common issues:
  - Missing Patches and Outdated Signatures
  - User Reluctance to Update Software

Supply chain vulnerabilities

End-of-Life Alerts: Monitor vendor announcements for product lifecycle changes to prevent security gaps.

Lifecycle Stages: Track end-of-sale, support, and life stages for timely IT product updates.

Embedded & Cloud Risks: Be vigilant of vulnerabilities in embedded systems and the risk transfer in cloud services.

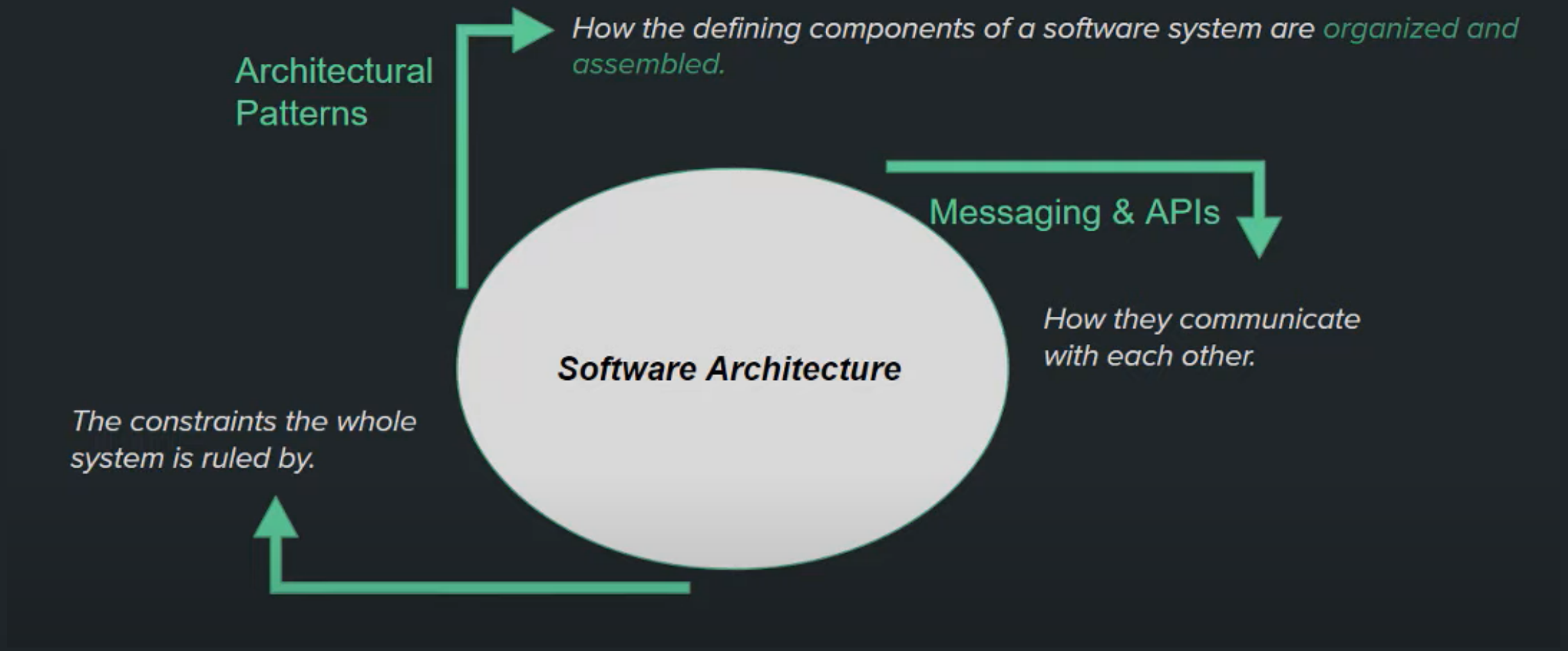
Vendor Oversight: Continuously evaluate vendor reliability and support for cloud and data storage services.

Configuration vulnerabilities

Avoid	default configurations
Verify	device security before network integration
Follow	documented security standards and best practices
Choose	secure cryptographic protocols and strong ciphers
Apply	the principle of least privilege

Architectural vulnerabilities

- Incorporate security early in IT system design to prevent fundamental flaws
- Address system sprawl with proper lifecycle management of networked devices
- Go beyond technical aspects to include business processes and human factors



Shift Left: Incorporating quality attributes requirement early and making them design criteria

Source: **Software Architecture | Architectural patterns | Architecture vs Design pattern**

Threat Modeling

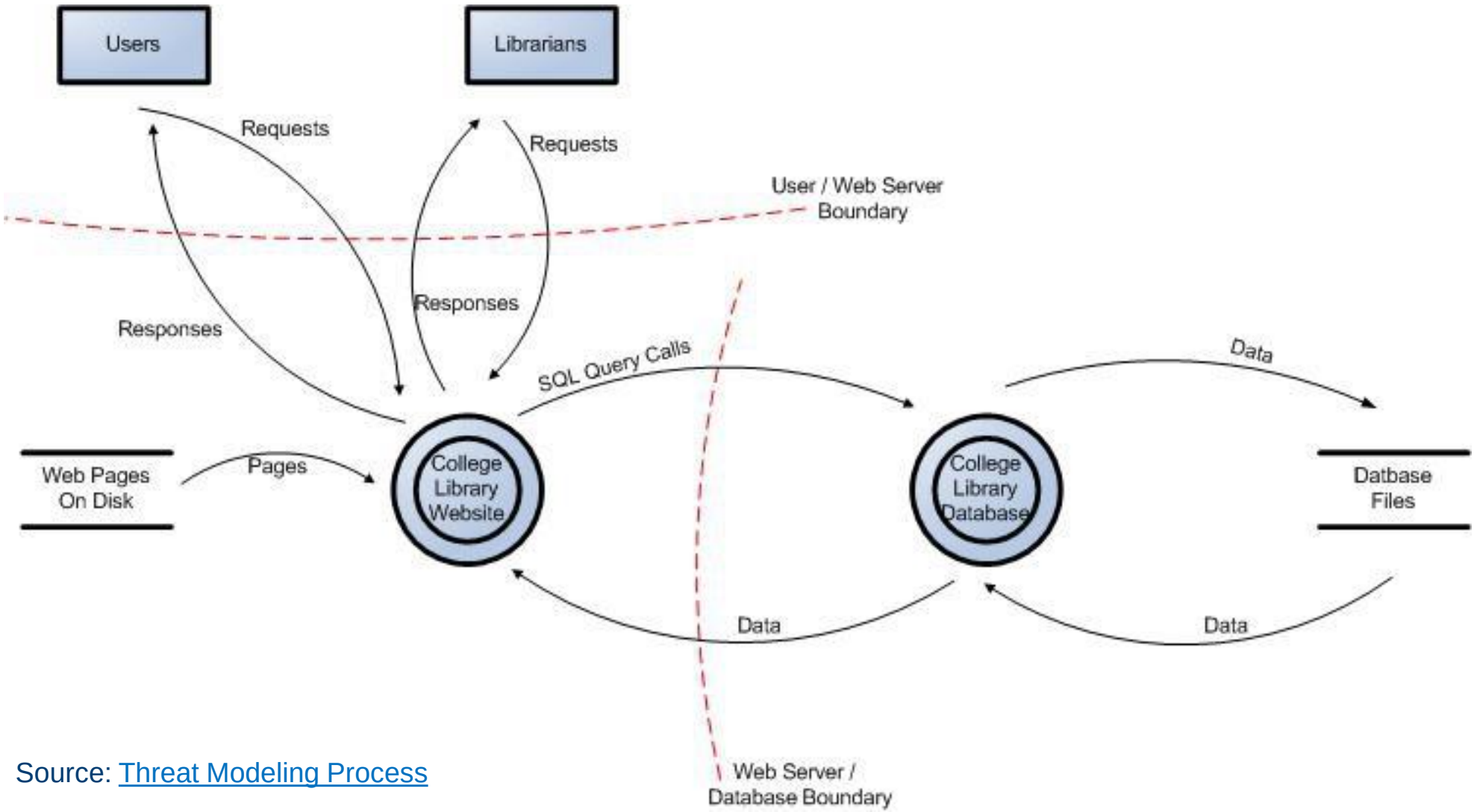
- Four questions framework
  - What are we building?
  - What can go wrong? (Be specific)
  - What are we going to do about that?
  - Did we do a good enough job?

STRIDE

Property	Threat	Definition
Authentication	**S**poofing	Impersonating something or someone else.
Integrity	**T**ampering	Modifying data or code
Non-repudiation	**R**epudiation	Claiming to have not performed an action.
Confidentiality	**I**nformation Disclosure	Exposing information to someone not authorized to see it
Availability	**D**enial of Service	Deny or degrade service to users
Authorization	**E**levation of Privilege	Gain capabilities without proper authorization

Source: https://www.microsoft.com/en-us/security/blog/2007/09/11/stride-chart/

Data Flow Diagram



Source: [Threat Modeling Process]

Vulnerability Management

Discover

Current-State Analysis

Prioritize

Roadmap for Program Execution

Validate

Measured Progress compared to Current State Analysis

Remediate

Program **Execution**

- Post Assessment Remediation
- Continuous Remediation
- Dedicated Remediation

Standards

- Security Content Automation Protocol (SCAP)
- CVE and NVD databases
- Common Vulnerability Scoring System (CVSS)

CVSS v2.0 Ratings	
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

CVSS v3.0 Ratings	
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Video: SCAP

Video: CVSS