CPSC 4970 Applied Cyber Security

AUBURN UNIVERSITY

Module 1

Welcome! CPSC 4970 Applied Cyber Security

AUBURN

- Understanding of the following concepts:
  - Security concepts related to software development lifecycle
  - Source code management techniques and workflows
  - Software composition and verification
  - Software vulnerabilities and verification techniques
  - Various security tools
  - Software build pipelines and automation
- Progressively build a software application build pipeline that incorporates cyber security practices.
  - Secure Software Development Life Cycle (SSDLC)
  - Secure Development LifeCycle (SDL)
- Hands on application of practices and tools used in modern software development organizations

Why is Cyber Security Important

AUBURN

2020 — **2021** — **2022** →

solarwinds Sunburst

CVE-2021-35211

Kaseya

CVE-2022-21660

Vue.js

CVE-2021-30116

LOG4J

CVE-2021-44228

LOG4J

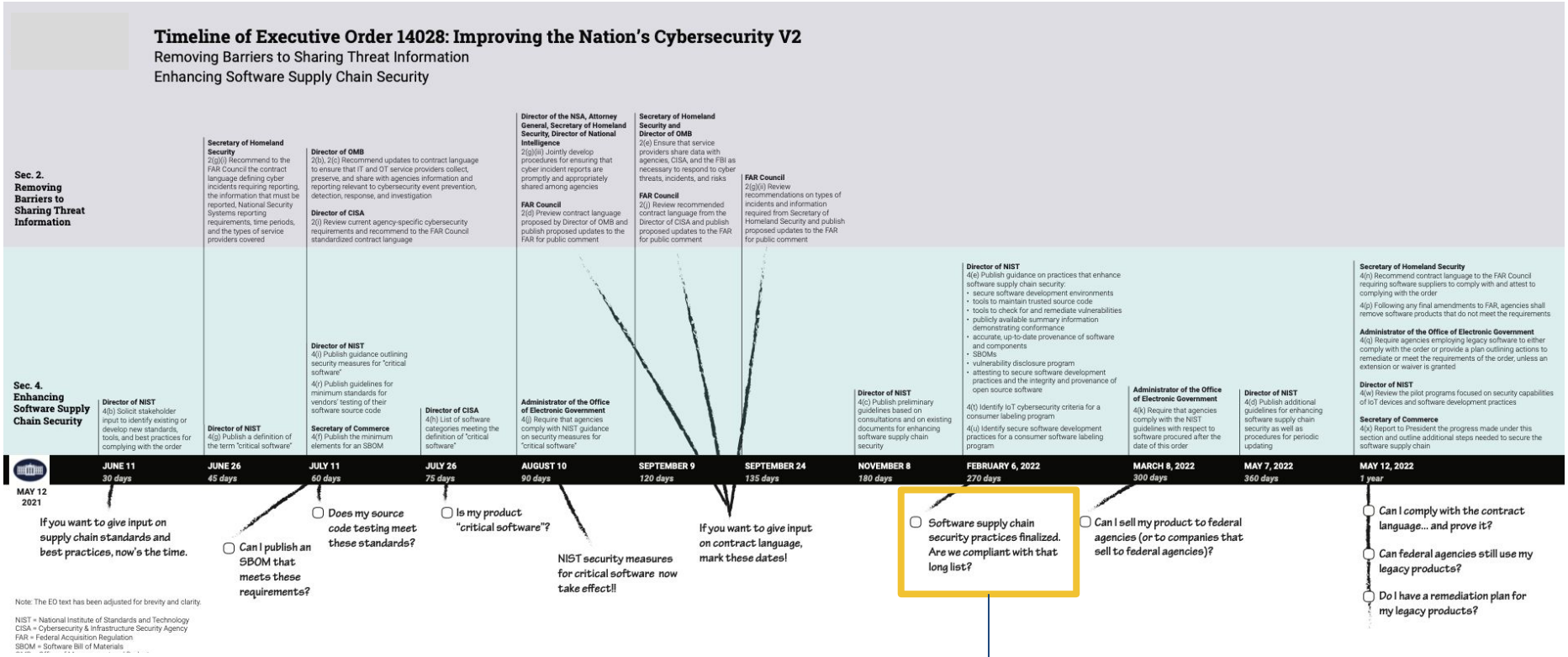
CVE-2021-45046

White House Drives Legislation

AUBURN

- Executive Order (EO) 14028 - May 12, 2021
  - "Improving the Nation's Cybersecurity" requiring the government to only purchase software that is developed securely.
  - Sec. 4 - *"Enhancing Software Supply Chain Security"* - The development of commercial software often lacks transparency, sufficient focus on the ability of the software to resist attack, and adequate controls to prevent tampering by malicious actors. There is a pressing need to implement more rigorous and predictable mechanisms for ensuring that products function securely, and as intended.
- July 28, 2021 - National Security Memorandum on Improving Cybersecurity for Critical Infrastructure Control Systems
- Memorandum M-21-30 - Aug 10, 2021 - Protecting Critical Software Through Enhanced Security Measures
  - Software that controls access to data, cloud-based and hybrid software, software development tools, such as code repository systems, testing software, integration software, packaging software, and deployment software, software components in operational technology (OT).

White House Drives Legislation



Improve Software Supply Chain Security

Improve Software Supply Chain Security

AUBURN

- Establish baseline security standards for development of software sold to the government, including requiring developers to maintain greater visibility into their software and making security data publicly available.
- Establishes concurrent public-private process to develop new and innovative approaches to secure software development and uses the power of Federal procurement to incentivize the market
- Creates a pilot program to create “energy star” type of label so government and public at large can quickly determine whether software was developed securely.
- Focuses on the using the purchasing power of the Federal Government to drive the market to build security into all software from the ground up.

Securing Software Dev Environments

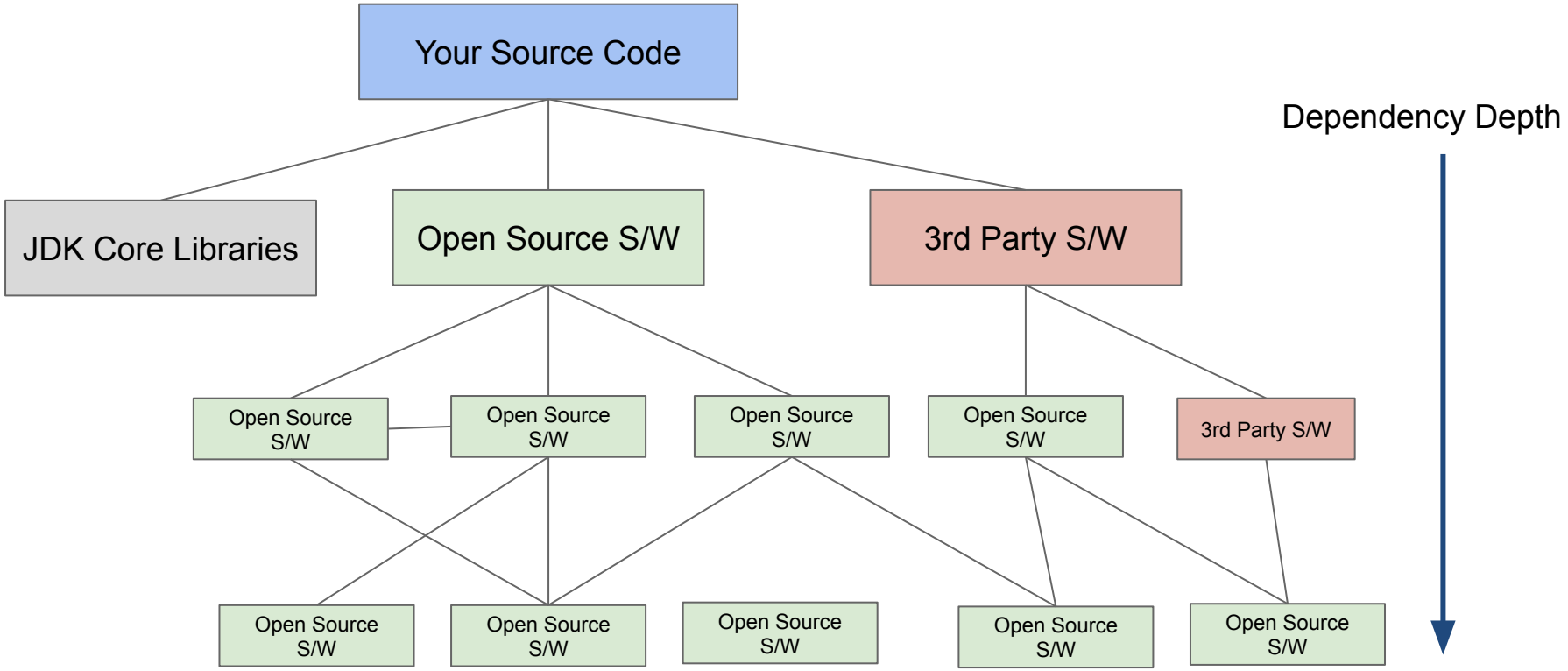
AUBURN

- Separate build environments with administrative controls
- Regular audits of access controls; implement advanced authentication mechanisms (multi factor).
- Employ data encryption
- Employing automated tools with access to trusted source code supply chains, thereby maintaining code integrity
- Automated tools to check for known and potential vulnerabilities to support quick action for remediation or risk mitigation.
- Provide proof of origin of software code or components and controls on internal and 3rd party software components, tools, and services present during development process.
- Perform audits on effectiveness of controls on a recurring basis.
- Coming: Software Bill of Materials - what does your software contain?

Software Composition

AUBURN

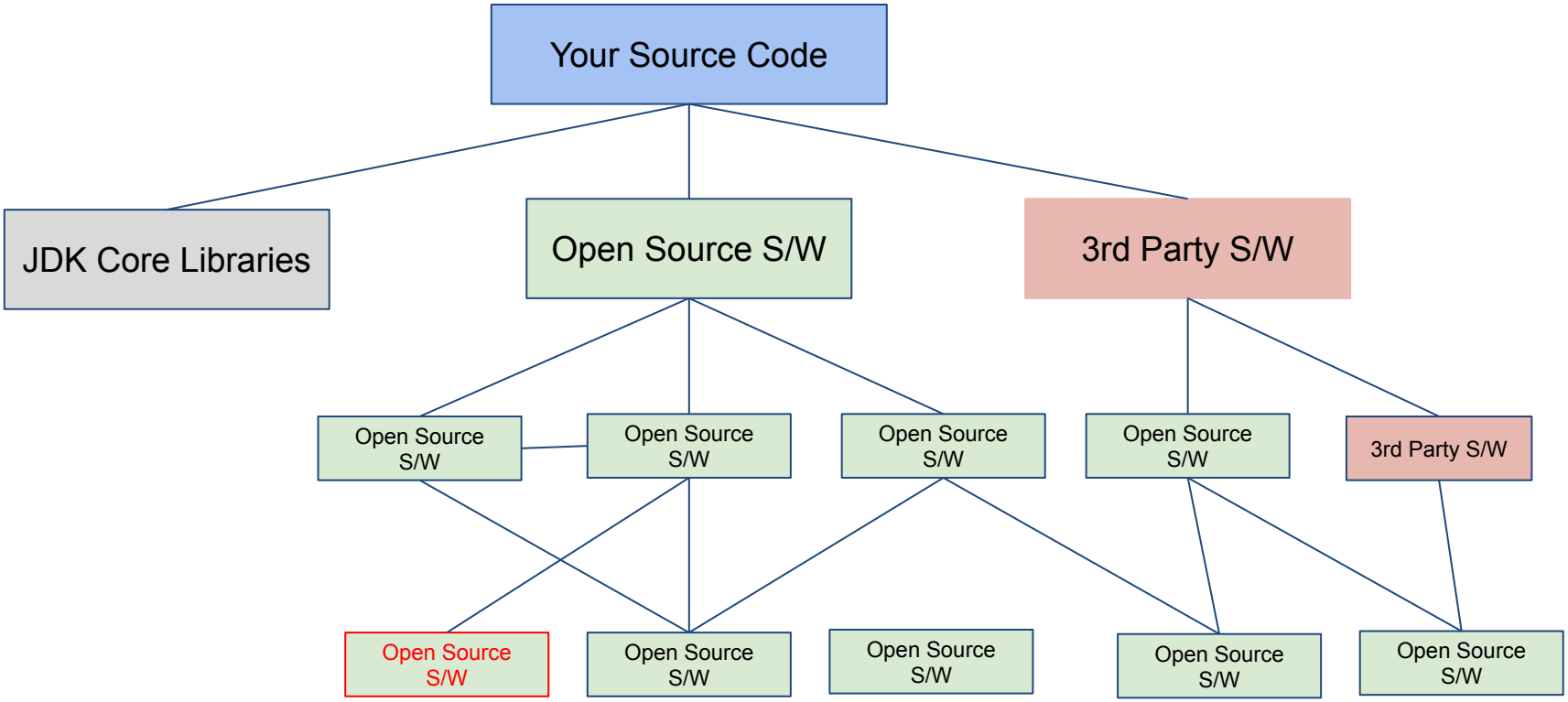
- Most application software utilizes larger amounts of open source and 3rd party software
- Not uncommon for application source code to be <25% of over all source code.



Software Composition Challenges

AUBURN

- How do you validate your developers are writing secure source code?
- How do you control the introduction of open source and 3rd party software?
- How are you notified when a vulnerability is uncovered in a software dependency?



Confidentiality, Integrity, Availability (CIA)

AUBURN

3 goals of secure software development to insure information security:

Confidentiality - Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information

Integrity - Guarding against improper information modifications or destruction, and includes ensuring information non-repudiation and authenticity

Availability - Ensuring timely and reliable access to and use of information.

Cyber Security Challenges

AUBURN

- Software security does not equal software quality.
- No point solution will provide a single solution for software security.
- A holistic defense-in-depth approach is required
- A blend of people, process, and technology. The most important part being people.
- This course will apply techniques utilizing people, process and technology to build a secure software development process

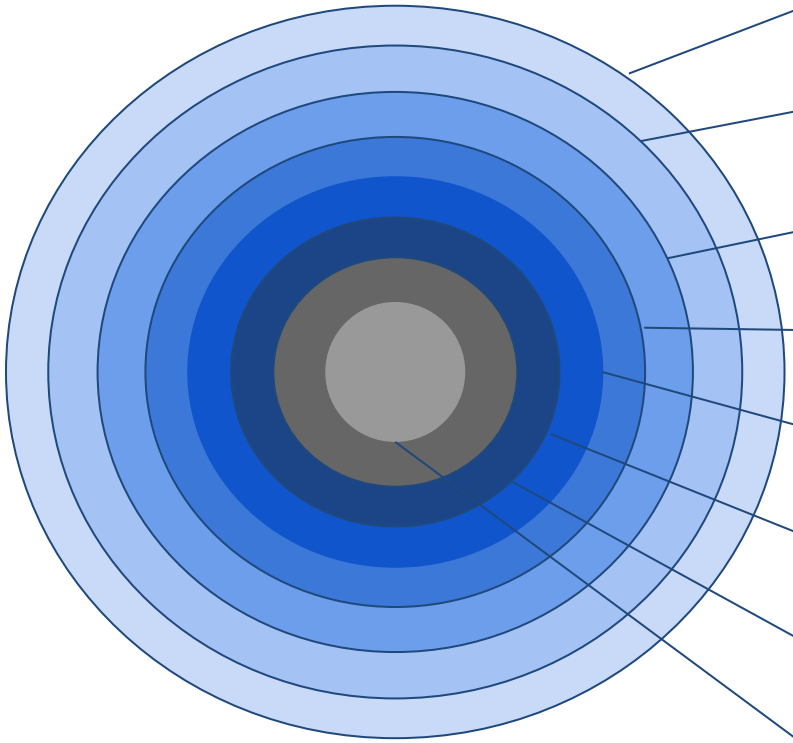
Defense in Depth Approach

AUBURN

- Historically information security industry has focused on network and application security, assuming software is secure and their security controls were sufficient protection.
- Security practices during the software development cycle is now considered a first step in securing software systems and reducing risk.
- **Defense-in-Depth** security approaches see the software development process as a essential part of policies and standards of a comprehensive cyber security system.
  - Defense-in-depth is a strategy that provides multiple security controls in case other security control fails or a vulnerability is exploited.
  - Originated from a military strategy by the same name, which seeks to delay the advance of an attack by constructing multiple defensive lines.
  - Defense-in-depth cybersecurity use cases include end-user security, product development, network security, etc.

Zero Trust Defense-in-depth approach

AUBURN



Client/Mobile Protection

Physical Equipment & Infrastructure

Perimeter Infrastructure

Internal Network

Compute

Operating System

Application Development & Deployment

Course Focus

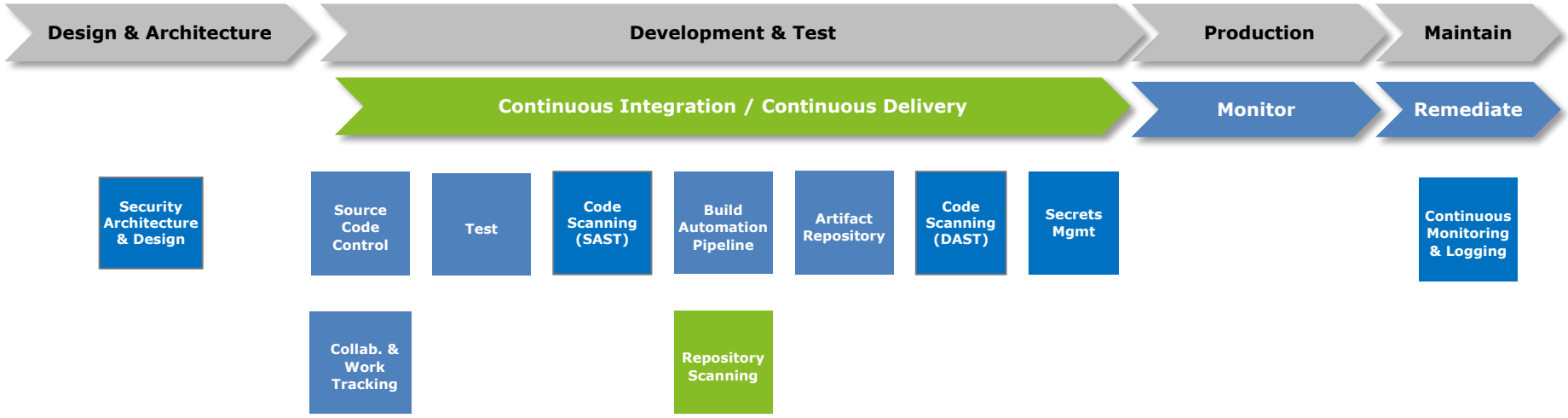
Data

Shift Left on Software Security

AUBURN

Shift Left - Integrating information security practices into daily work, software teams can achieve higher levels of software delivery performance and build more secure systems. Security concerns, are addressed earlier in the software development lifecycle (that is, left in a left-to-right schedule diagram).

Course Objectives & Project



Security Automation

The following are commonly identified security gaps in a DevOps Pipeline:

- Static code scanning to support code quality and secure coding standards
- Dynamic application security testing for applications is performed
- Continuous logging and monitoring of production environment

Manual Security Methods

Checkpoints and gates must be implemented so security controls are in place:

- Security and architecture based on attack surfaces and data sensitivity.
- Ongoing source code change management controls based on code inspections and reviews.
- Periodic penetration tests are performed (e.g., OWASP Zap)