

# Secure DevOps: Application Security Principles and Practices

## Introductions

Your name  
Your Title  
Microsoft



# Course outline

- Module 1: Evolution to Secure DevOps
- Module 2: Secure DevOps Principles and Practices
- Module 3: Application Security Principles
- Module 4: Automating a Secure and Compliant Pipeline
- Module 5: Threat Modeling Concepts
- Module 6: Manual Security Verification

# Introduction

Your role

Your experience in Secure DevOps

Your goals for this workshop

# Workshop context

# Microsoft Azure Well-Architected Framework

Architecture guidance and best practices, created for architects, developers and solution owners, to improve the quality of their workloads, based on 5 aligned and connected pillars

**Cost  
Optimization**



**Operational  
Excellence**



**Performance  
Efficiency**



**Reliability**

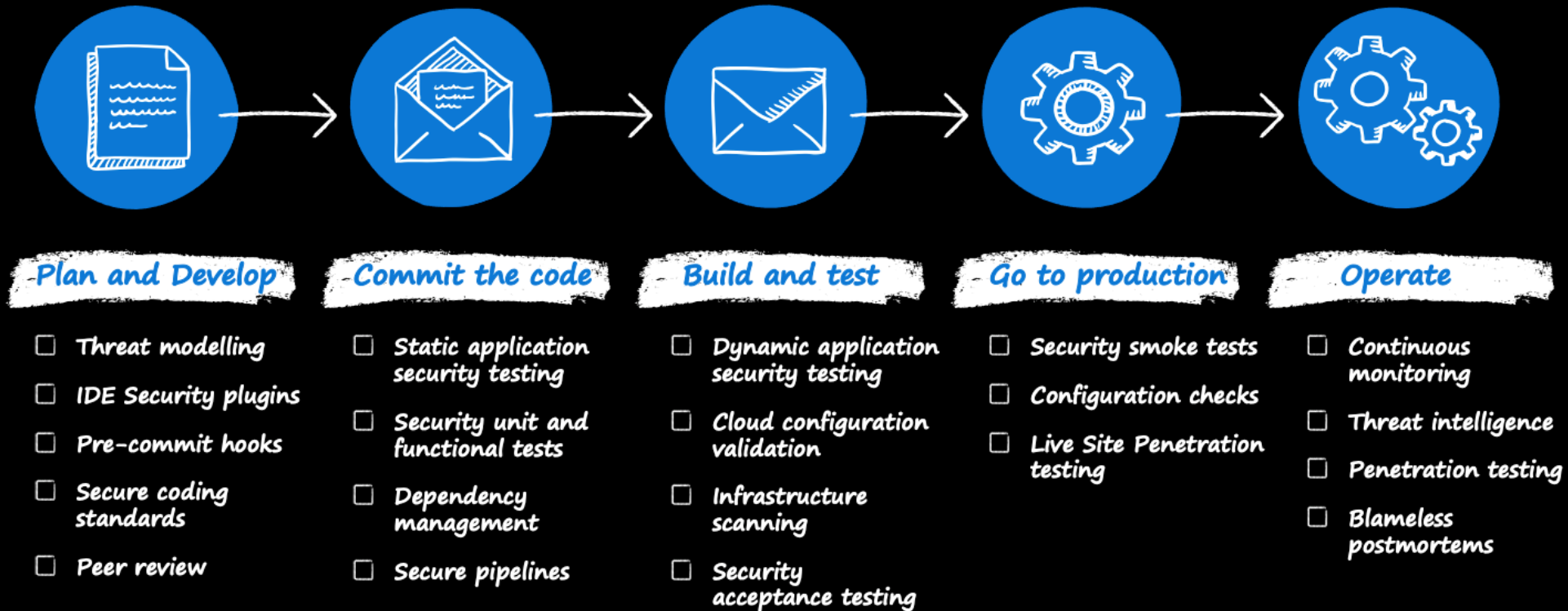


**Security**



<https://aka.ms/wellarchitected/framework>

# Cloud Adoption Framework (CAF)



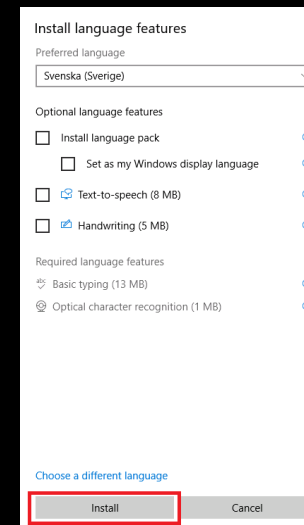
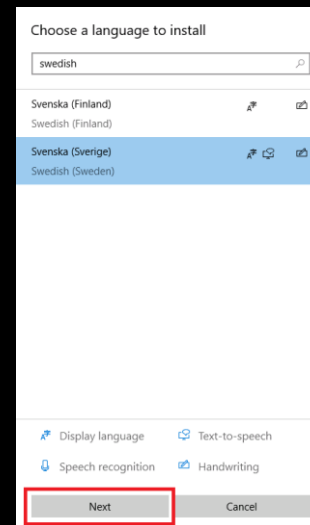
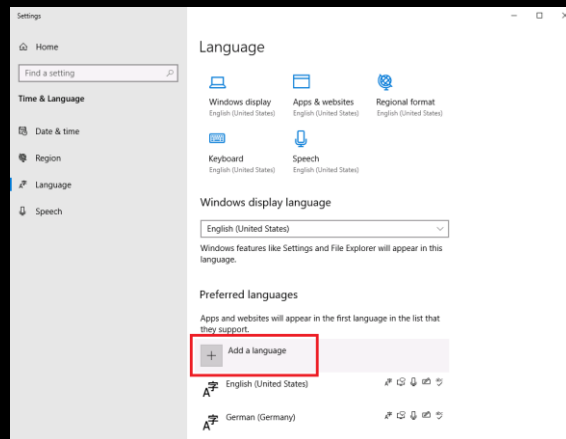
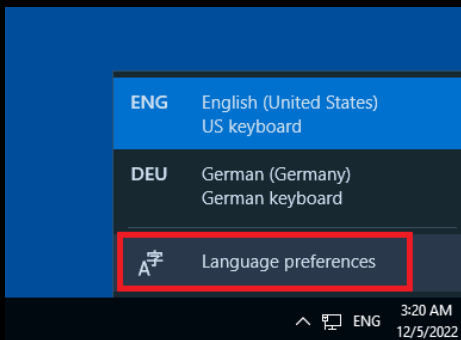
# Lab access

<http://aka.ms/LOD>

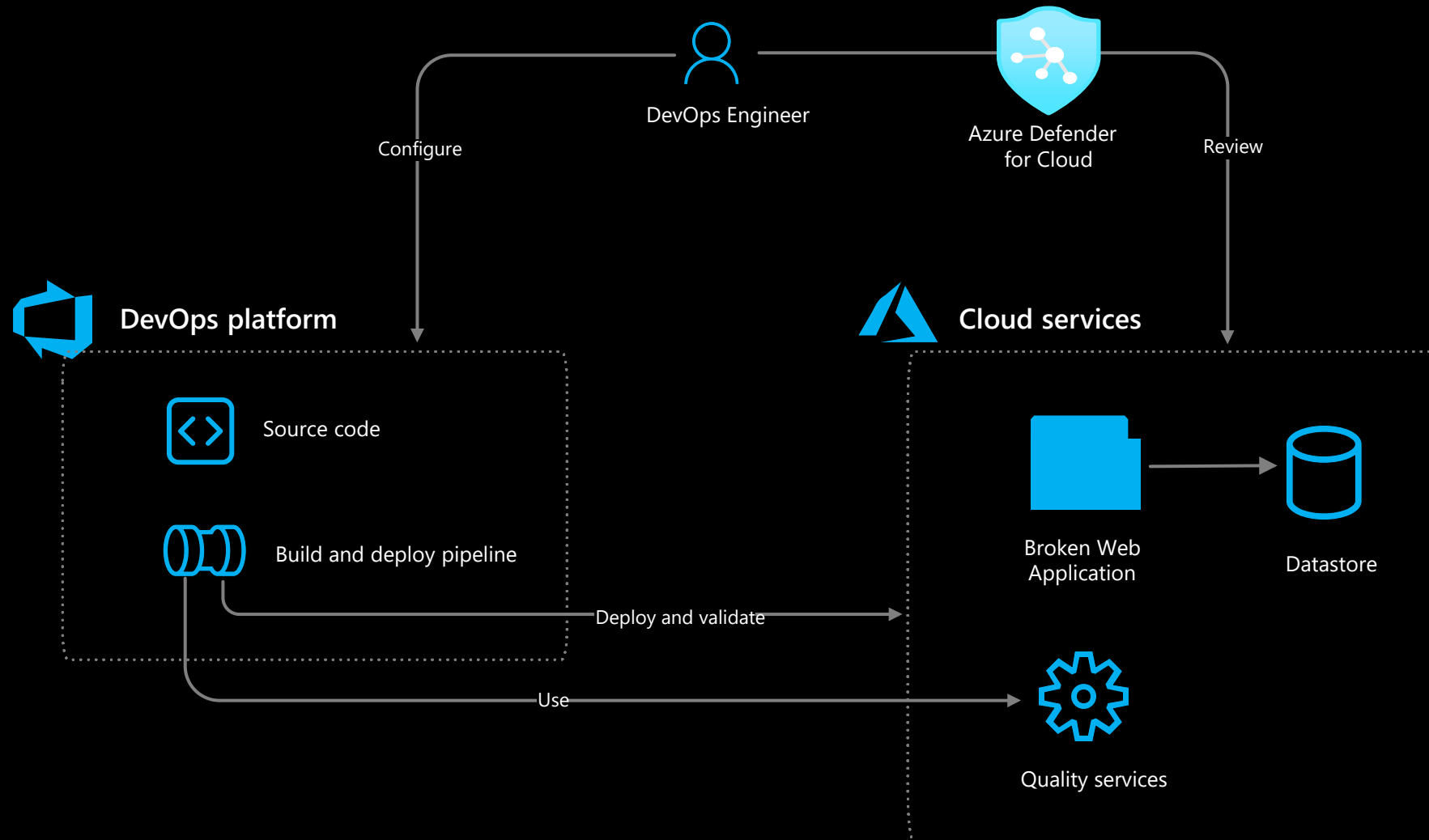
Training Key: **C944B795D956406E**

# Lab setup

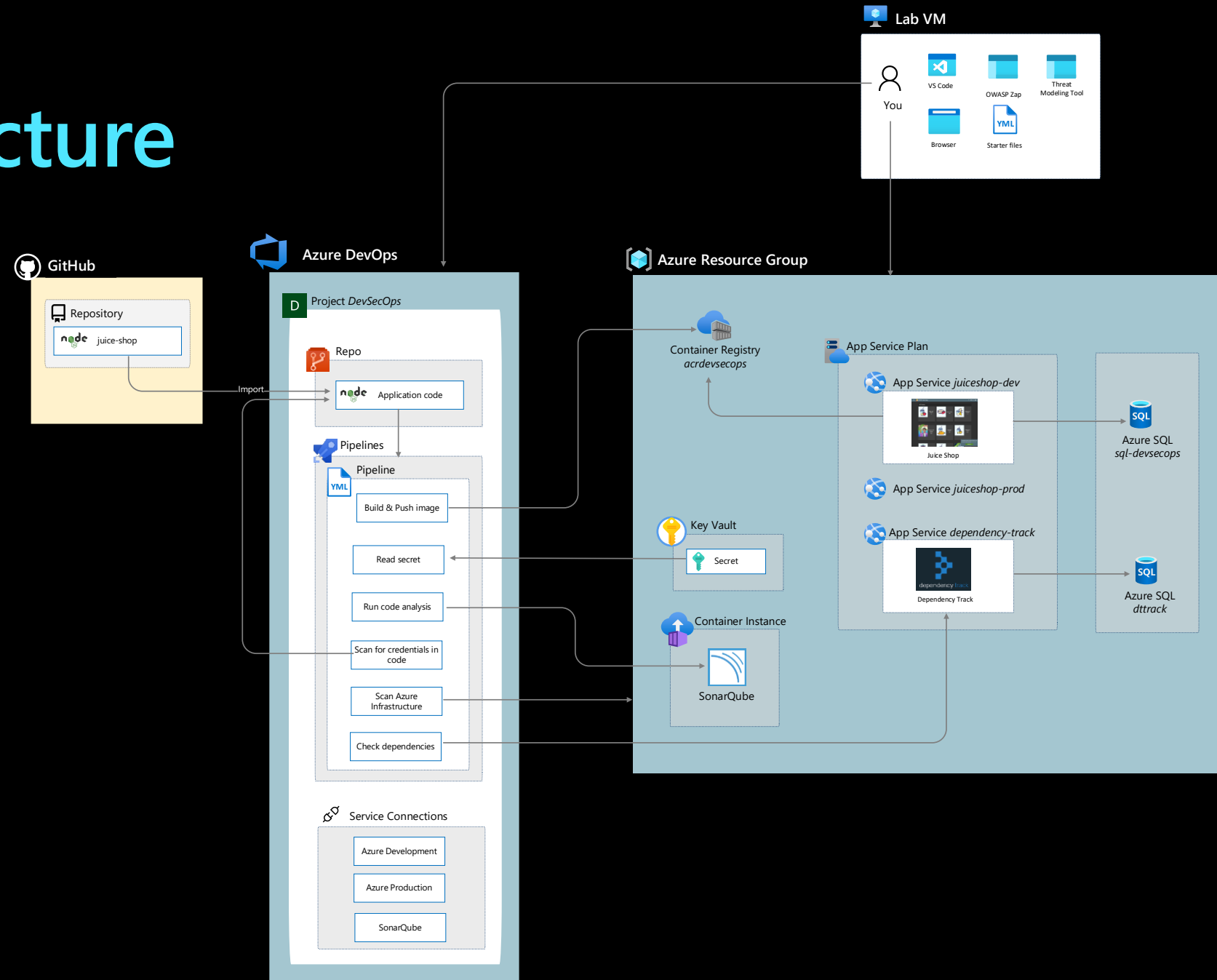
## Switch keyboard language



# Lab scenario



# Lab Architecture



# Acronyms

**IOC:** Indicator of compromise

**STIX :** Structured Threat Information eXpression

**TAXII:** Trusted Automated eXchange of Indicator Information

**RASP:** Runtime Application Self-Protection

**UEBA:** User and Event Behavioral Analytics

**IAST:** Interactive application security testing

**SAST:** Static application security testing

**SVT:** Security Verification Testing

**DAST:** Dynamic application security testing

**SCA:** Software Composition Analysis

**NVD:** National Vulnerability Database

**SCA:** Software Composition Analysis

**SSL:** Secure Sockets Layer

**GDPR:**

**OSA:** Operational Security Assurance

**CWE:** Common Weakness Enumeration

**OAuth2 :**

**OIDC:** Open ID Connect

**CVE:** Common Vulnerabilities and Exposures

**SIEM:** Security Information and Event Management

**SOAR:** Security Orchestration, Automation and Response

**CSPM:** Cloud Security Posture Management

**CWPP:** Cloud Workload Protection Platform

**CIA:** Confidentiality, Integrity, Availability

**STRIDE:** Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege

**MSRC** Microsoft Security Response Center

**DREAD:** Damage, Reproducibility, Exploitability, Affected Users , Discoverability

**CSRF/XSRF:** Cross-site request forgery

**XSS:** Cross-site scripting

**WAF:** Well-Architected Framework

**CAF:** Cloud Adoption Framework

**NSA:** National Security Agency

**CIA:** Central Intelligence Agency

**CVSS:** Common Vulnerability Scoring System

**JNDI:** Java Naming and Directory Interface

**NIST:** National Institute of Standards and Technology

**C2:** Command and Control

**C&C:** Command and Control

**CVE:** Common Vulnerabilities and Exposures

**OWASP:** Open Web Application Security Project

**PKCE:** Proof Key for Code Exchange

**DFD:** Data Flow Diagram

**MTTC:** Mean Time to Compromise

**MTTP:** Mean Time to Privilege escalation

**MTTE:** Meat Time to Exfiltration

**TTP:** Tactic, Tools and Procedures

**MSRC:** Microsoft Security Response Center

**OSA:** Operational Security Assurance

# Definitions

**Software Vulnerability:** A security flaw, glitch, or weakness found in software code that could be exploited by an attacker (threat source)

**Attack Surface:** Any part of an application that is accessible by a human or another program

**Attack Surface Reduction:** Minimize the number of exposed attack surface points a malicious user can discover and attempt to exploit

**Privacy:** Empowering users to control collection, use, and distribution of their personal information

**Security:** Establishing protective measures that defend against hostile acts

**Zero Day:** Malicious actors knew about vulnerability and exploited it before experts did

**Trust Boundary:**

**Trust boundary violation:**

**Software supply chain:** Dependencies and integration of any third-party or open-source software.

**Command and control:** The process through which an attacker establishes a connection with a compromised asset that they have taken control of in a target network

**CVE:** a database of publicly disclosed information security issues. Launched by MITRE organization

**CWE:** community-developed list of software and hardware weakness types

**Residual Risk:** the amount of risk or danger associated with an action or event remaining after natural or inherent risks have been reduced by risk controls

# Tools

**Whitesource/Mend**

**SonarQube**

Azure Tenant Security Solution (AzTS)

CodeQL

Microsoft DevSkim

BinSkim

Attack Surface Analyzer

GitHub Dependabot

GitHub Advisory Database

GitHub Dependency Graph

GitHub Code scanning

GitHub Secret scanning

PREfast

FXcop

MiniFuzz

# Example attacks and vulnerabilities

**Love Bug:** 2000

**Nimda:** 2001

**CodeRed:** 2001

**Log4j:** 2021

**SolarWinds:** 2020

**Equifax:** 2017, database breach PII information. Leveraged vulnerability in Apache Struts

**Heartbleed:** OpenSSL

**Struts:**

**WannaCry:** 2017, ransomware, 300,000 computers[ across 150 countries