Secure SDLC

ADAM AIRHART
RYAN HEJNOSZ
BRETT LESCZYNSKI

Security embedded in all steps of the process

- Plan
- Code
- Build
- Test
- Release
- Deploy
- Monitor
- Operate

←SHIFT LEFT

Security must be easy to implement as early in all process as possible

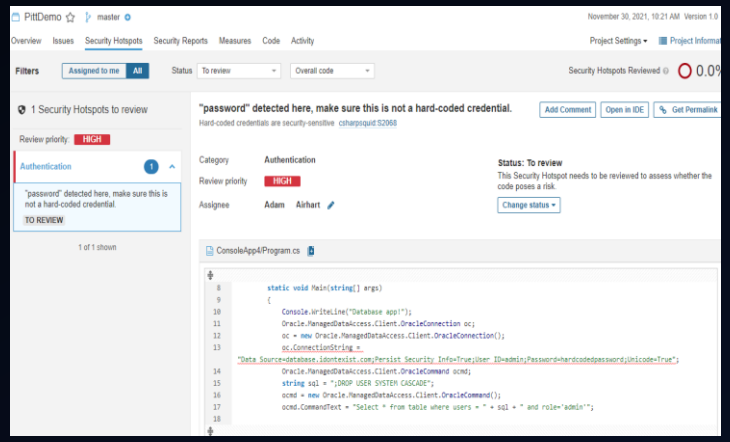
- Plan
- Code
- Build
- Test
- Release
- Deploy
- Monitor
- Operate

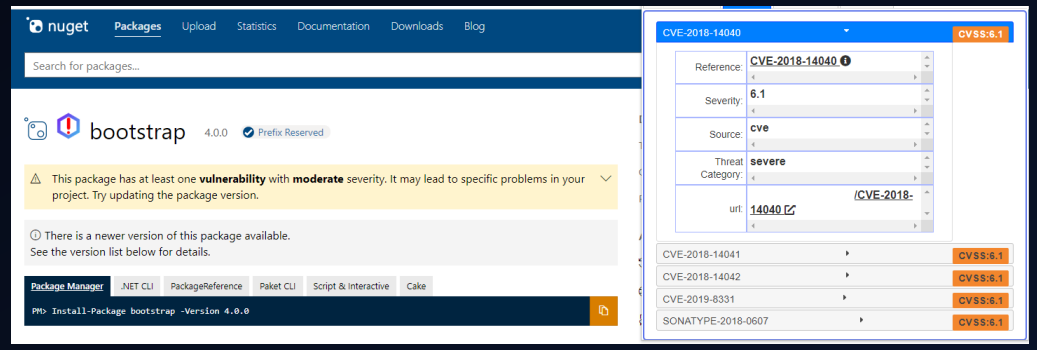
Plan

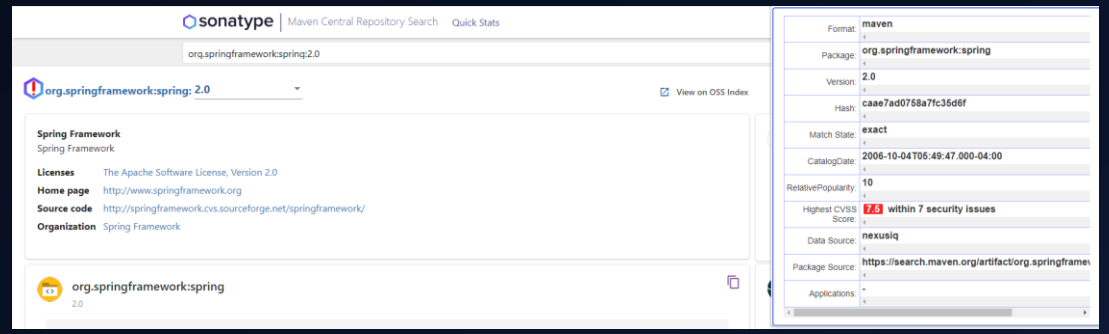
- Specifications
  - Business requirements
- Identify technology
  - .NET/Java
  - Tomcat, IIS
  - Container?
- Time lines/deadlines
  - Security should not be an after thought to meet a deadline

Code

- Code Analysis
  - OWASP Top 10
  - CWE Top 20
  - SANS Top 25
- Secure coding guidelines
  - Provide training on secure developent
- Identify Open Source packages
  - Check for CVEs
  - Check for licensing compliance







Code Scan Demo

Build

- Automated Builds
  - Scan every build
    - Same scans as code phase, safety net
  - Break on policy failure
    - Non compliant code cannot be installed
    - Common violations, provide training

Test

- Difficult with old applications
  - Not written with testing in mind
  - Older technologies
  - Missing business knowledge
- Easy with new applications
  - Automated testing
  - Test driven coding
  - Frameworks available

Release

- Workflow
  - Release to test
    - TEST
  - Release to QA
    - TEST AGAIN
  - Release to Production
    - Verify previous tests
- Package
  - All artifacts, work items, approvals, test results

Deploy

- Approvals
  - Developer who wrote change cannot deploy change
- Dynamic Application Security Testing
  - Automated “pen test lite”
    - OWASP Top 10
    - Remote Code Execution
- Phases
  - Test Deployment
  - QA Deployment
  - Production Deployment

Monitor/Operate

- Review logs
  - Abnormal security event?
  - Access from unusual IP
  - Elevated privileges
- Infrastructure
  - Is system logging?
  - Are dependencies up?
  - CPU, RAM, HDD utilization
  - OS Event logs
- Vulnerability Management
  - System patching
  - Component updates and patching

Questions?