EDUCAUSE

Wednesday, May 3rd

SANS

Taking Control of Your Application Security

INTRODUCTION

Eric Johnson, CISSP, GSSP-Java, GSSP-.NET, GWAPT

- Application Security Curriculum Manager, SANS Institute
  - SANS Certified Instructor, Author
- Senior Security Consultant, Cypress Data Defense
  - Security assessments – source code reviews, web app pen tests, mobile app pen tests
  - Coder – security tools, demos, not enough to be called a developer anymore
- Iowa State alum
  - B.S. Computer Engineering, M.S. Information Assurance
- Contact information
  - ejohnson@sans.org
  - @emjohn20

Agenda

- Taking Control of Your Application Security

TAKING CONTROL OF YOUR APPSEC

1. *The Threat Landscape*
   - **Talent Shortage**
2. **Taking Control**
   - **AppSec Ninja Program**
   - **Awareness Training**
   - **Secure Coding**
   - **Shifting Security Left**
   - **Dependency Management**

Confirmed data breaches by incident classification:

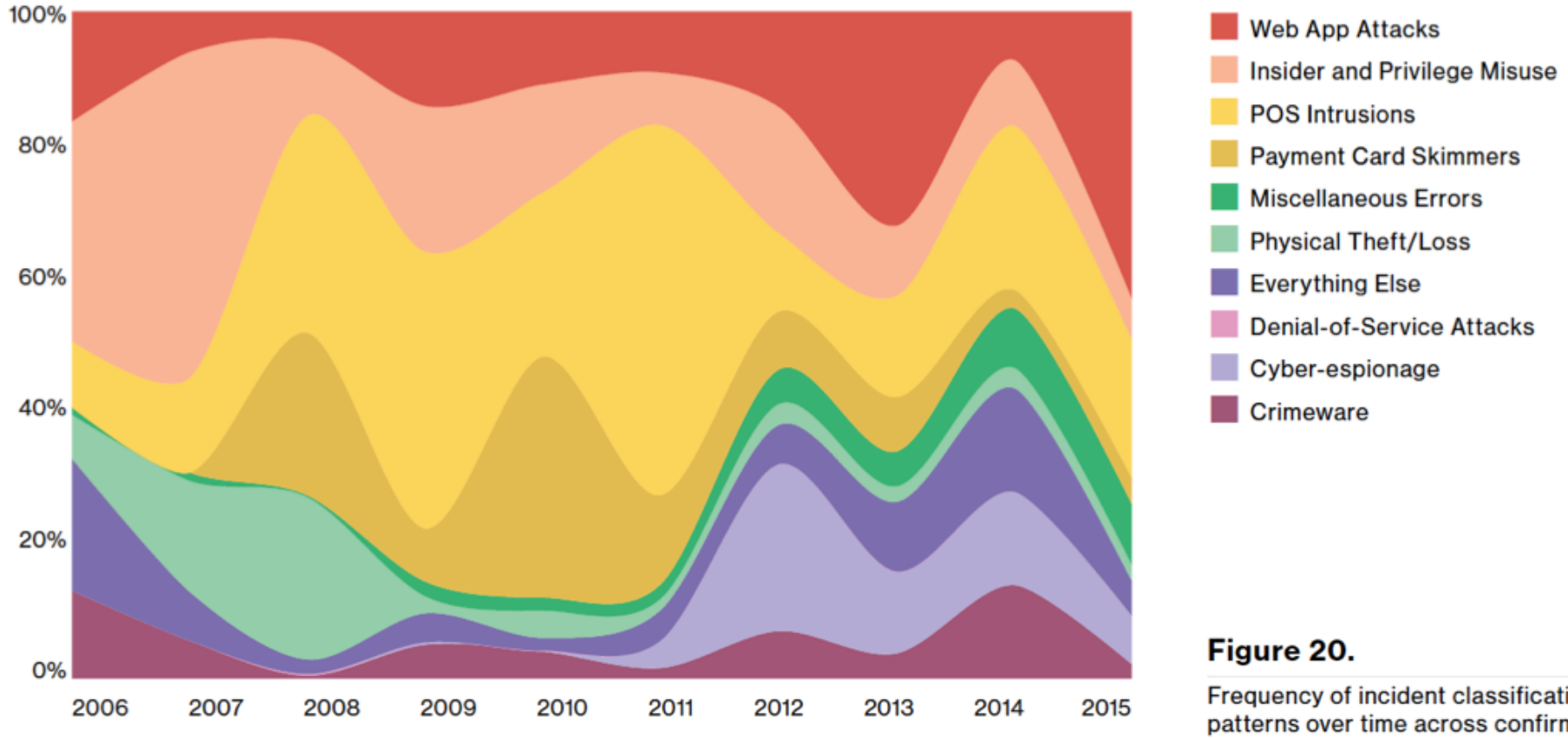
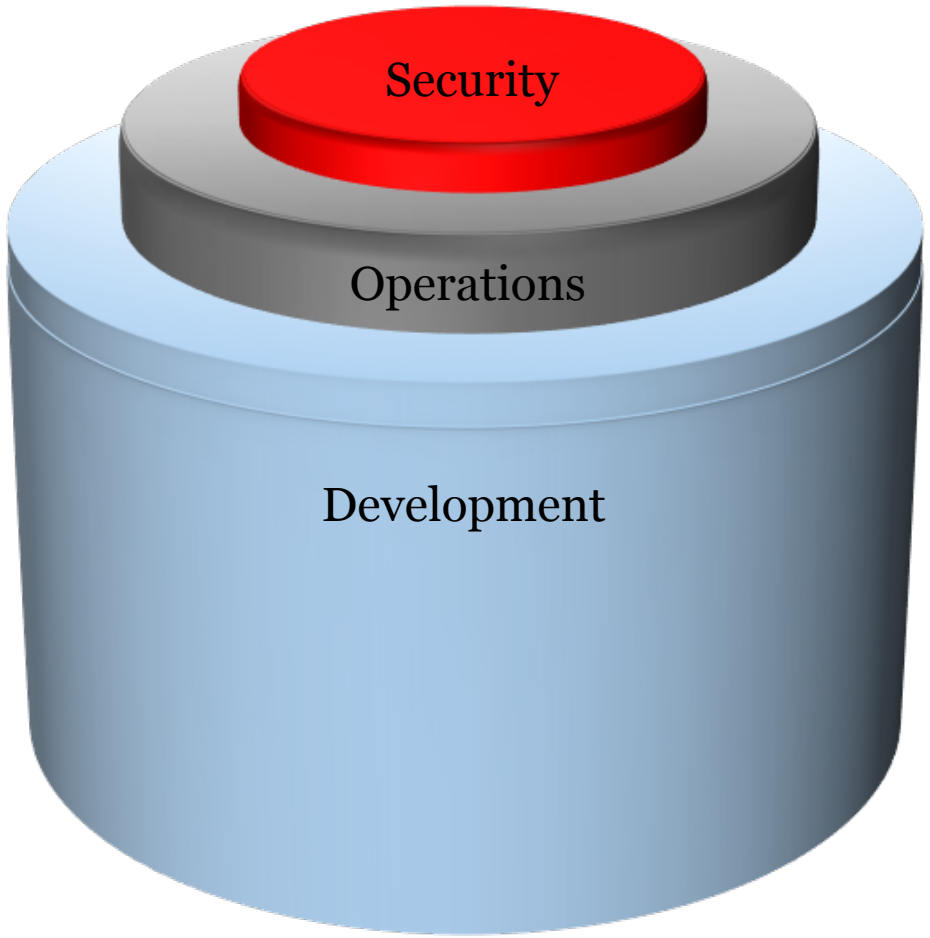


Figure 20.

Frequency of incident classification patterns over time across confirmed data breaches.

HERE IS THE PROBLEM

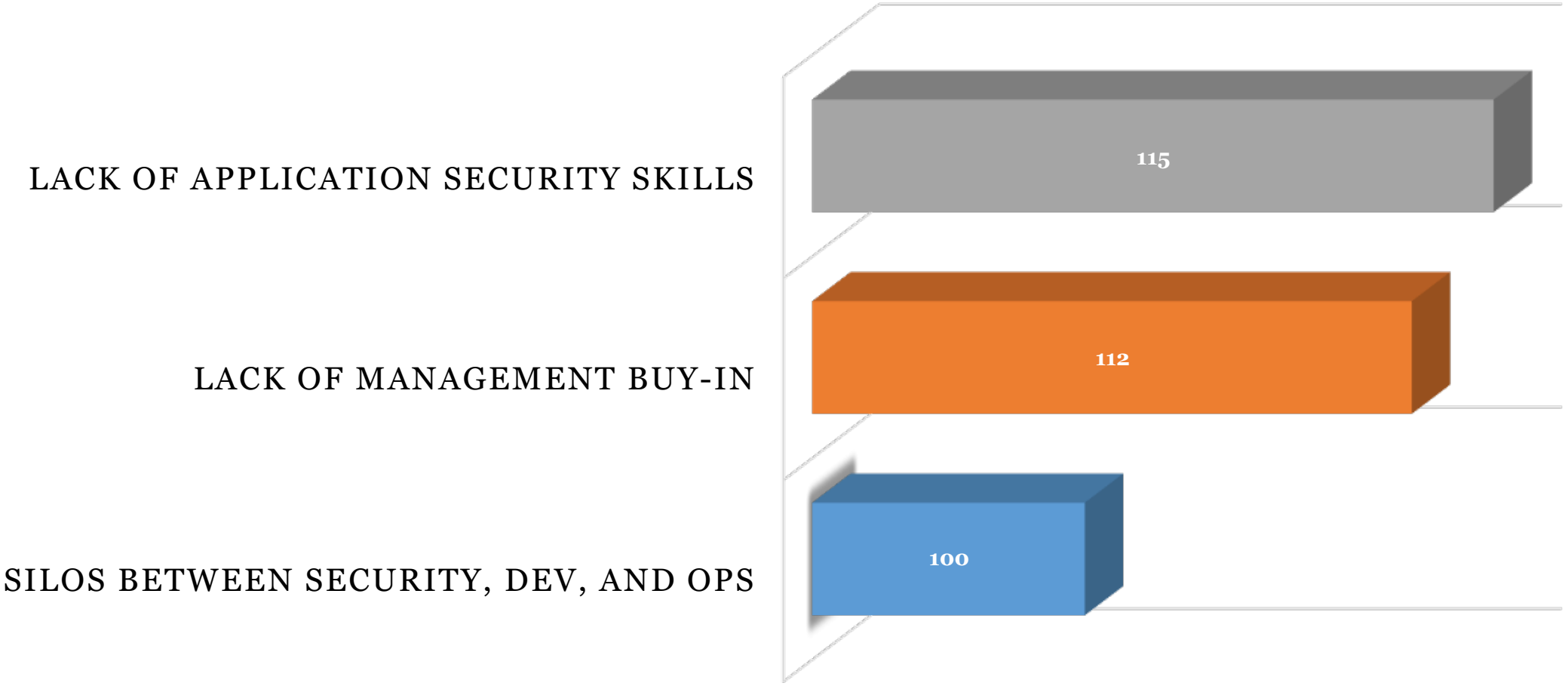
- Shortage of talented app sec engineers
- Building Security in Maturity Model (BSIMM) Version 7:
  - 2016 survey of 95 different firms from 6 different verticals including financial, software vendors, cloud, healthcare, IoT, and insurance
- Dev / Sec Ratio :
  - 245 Developers
  - 1 Software Security Expert



https://go.bsimm.com/hubfs/BSIMM/BSIMM7.pdf

2016 SANS APPLICATION SECURITY SURVEY

300 respondent's top challenges implementing an app sec program:



TEACHING SECURITY FROM THE BEGINNING

Quote from CSO Online:

"It is going to take 20 years to get where we need to be in having schools at all levels have security teachers and professors teach security to students so the workforce is stacked with security experts."

Jason Hoffman

Chief Security Officer, Marketo

Agenda

- Taking Control of Your Application Security

TAKING CONTROL OF YOUR APPSEC

1. The Threat Landscape
   - Talent Shortage
2. *Taking Control*
   - AppSec Ninja Program
   - Awareness Training
   - Secure Coding
   - Shifting Security Left
   - Dependency Management

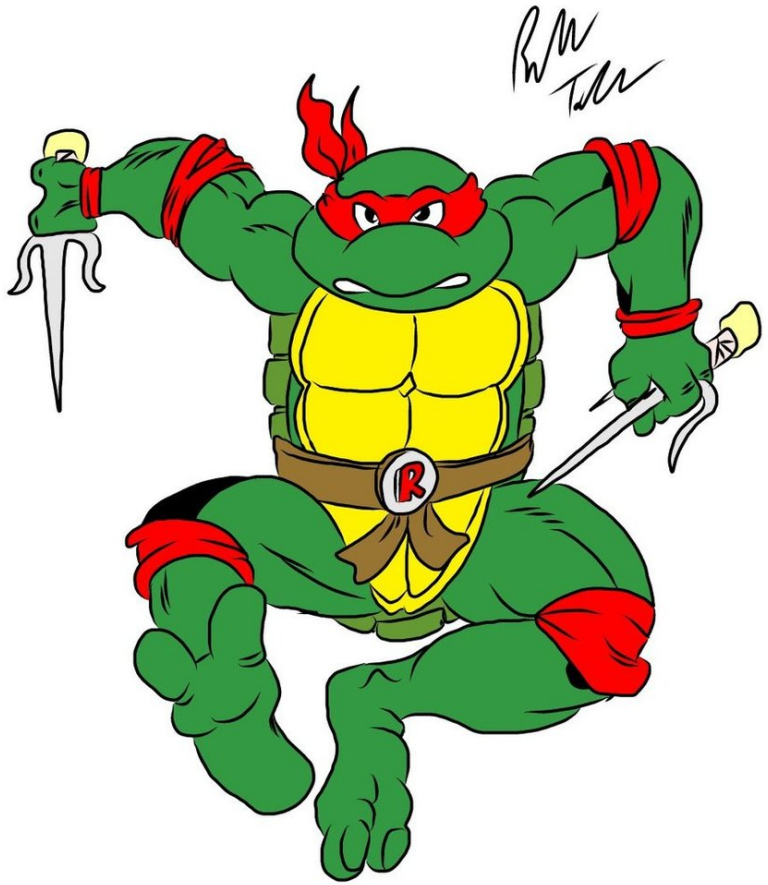
HELP WANTED: APPSEC NINJA CHECKLIST

Find the right mindset:

- ❑ Eagerness to learn
- ❑ Positive attitude
- ❑ Resourceful

Recommended technical skills:

- ❑ Proficient in organization's programming languages
- ❑ Knowledge of web & mobile technologies
- ❑ Familiar with modern development practices (DevOps toolchain)



img: th00.deviantart.net/fs70/PRE/i/2011/126/1/c/raphael_the_ninja_turtle_by_outsiderinbrianza-d3fp6vz.jpg

APPSEC NINJA PROGRAM: MANAGEMENT CHECKLIST

Management Checklist:

- ❑ Provide the right tools for the job
  - Static, dynamic, dependency management, DevOps toolchain
- ❑ Invest in people
  - Training & certification
  - Rotations between Dev and AppSec
- ❑ Build a sustaining culture
  - Assign an experienced mentor to trainees
  - Learn, present, and demo a new vulnerability to the mentor and development staff every week

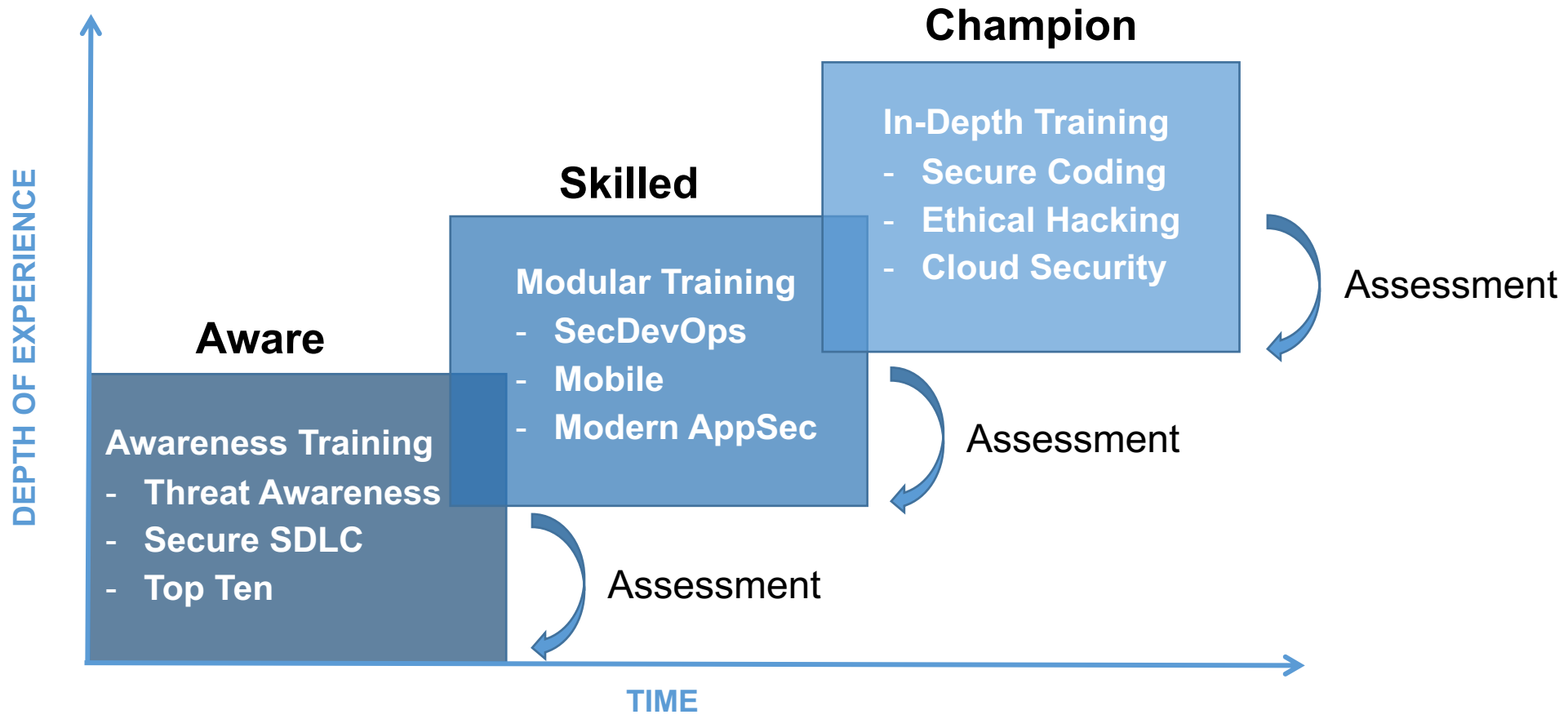
APPSEC NINJA PROGRAM:TRAINEE CHECKLIST

Going from Dev to AppSec:

- ❑ Research framework security features
- ❑ Write security unit and functional test methods
- ❑ Participate in security-specific peer reviews
- ❑ Create reusable security libraries / frameworks
- ❑ Run security scanning tools
- ❑ Verify scan results and filter false positives
- ❑ Attend conferences to network and share ideas

APPSEC NINJA SKILL LEVELS

Going from software engineer to security champion:



APPSEC INDEPTH TRAINING CHECKLIST

Instructor led, exercise driven, multiple day application security training:

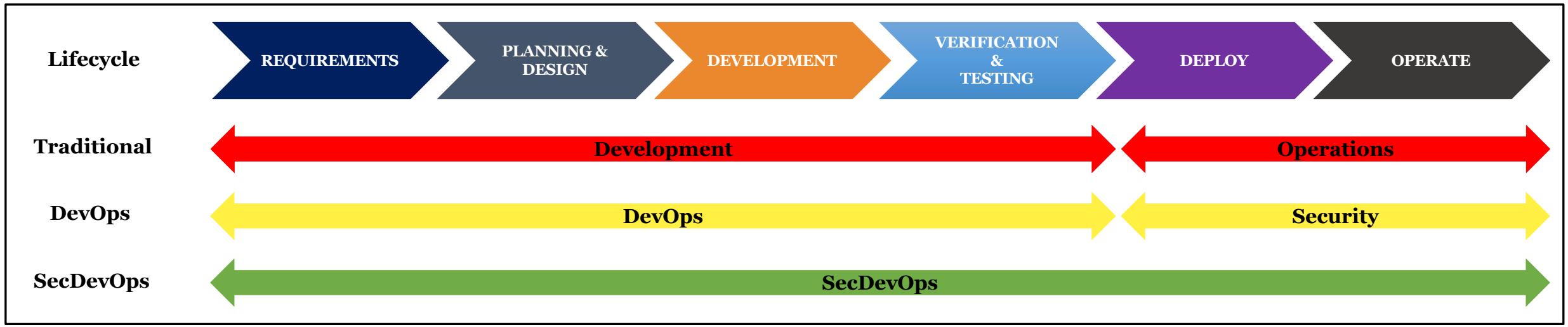
- ❑ Web app security
- ❑ Mobile app security
- ❑ Secure coding (Java, dotNET, C, C++)
- ❑ SecDevOps
- ❑ Cloud security
- ❑ Modern (web 2.0) framework security

  SPAs, Node.js, HTML5, JWTs, React, AngularJS

SHIFTING SECURITY LEFT

Move security into ALL phases of the development lifecycle:

- Identify risks using threat modeling during planning
- Automate unit testing for security stories
- Iterative, incremental scans during code, test, and release



STATIC ANALYSIS & CODE REVIEW CHECKLIST

Identifies potential application vulnerabilities in the source code:

- ❑ Integrate static analysis into the developer's IDE
- ❑ Code commits trigger automated light-weight static scans to minimize false positives
- ❑ High-risk code changes trigger alerts and manual code reviews
- ❑ Build pipeline runs static scans that return pass / fail results
- ❑ Scan results are automatically fed into the development backlog
- ❑ Schedule out of band in-depth code reviews by the security team

DYNAMIC ANALYIS CHECKLIST

Identifies application vulnerabilities in the running application:

- ❑ Integrate dynamic scanning into the developer's IDE
- ❑ Create security abuse cases in test suites
- ❑ Leverage a scanning framework to execute dynamic scans for common vulnerabilities
- ❑ Include security testing in QA test plans
- ❑ Schedule out of band in-depth penetration tests by the security team

DEPENDENCY MANAGEMENT CHECKLIST

Running vulnerable dependencies in our environment makes our systems inherently vulnerable

- ❑ Inventory all apps, dependencies, and versions
- ❑ Monitor bug lists for vulnerabilities (MITRE CVE, NIST NVD)
- ❑ Integrate scans for vulnerable dependencies into the build pipeline
- ❑ Zero-day dependency incident response plan
  - Virtual patching (WAF, RASP)
  - Dependency patching

DEPENDENCY MANAGEMENT TOOLS

Security tools for vulnerable dependency scanning:

- Free / open source:
  - OWASP Dependency Check
  - Retire.js
- Commercial:
  - Sonatype
  - Black Duck
  - Palamida
  - Source Clear

CONTINUOUS MONITORING & FEEDBACK CHECKLIST

Leverage monitoring tools and approaches for security monitoring:

- ❑ Look for attack signatures
  - Authentication failures, 4XX/5XX errors, database syntax errors, login failures, access control exceptions
- ❑ Correlate with traffic information (source, type)
- ❑ Feed trends and anomalies back to monitoring tools (statsd, graphite, graphana)
- ❑ Must watch: Christopher Rimondi "Using DevOps Monitoring Tools to Increase Security Visibility"
  - https://www.youtube.com/watch?v=TNCVv9itQf4

CONTINUOUS MONITORING DASHBOARD – ETSY

SANS AppSec
APPLICATION & SOFTWARE SECURITY

Security Awareness Training for Developers

Topics and Modules

OWASP Top Ten

- Introduction
- Injection Flaws
- Authentication
- Session Management
- Cross-Site Scripting
- Insecure Direct Object Reference
- Security Misconfiguration
- Insecure Cryptographic Storage
- Insufficient Transport Layer Protection
- Missing Functional Level Access Control
- Cross-Site Request Forgery
- Using Known Vulnerable Components
- Unvalidated Redirects and Forwards

Threat Awareness
NEW!

- Business Case
- Understanding the Attacker
- The Attack Process
- Trust Nothing
- Threat Modeling

Software Development Lifecycle

- Waterfall Model
- Agile Development
- DevOps

Classic Issues

- Memory Inspection
- Buffer Overflow
- Improper Error Handling

Top Design Flaws
COMING IN 2017

- Defense in Depth
- Separation of Concerns
- Single Responsibility
- Least Knowledge
- Don't Repeat Yourself

SANS AppSec

APPLICATION & SOFTWARE SECURITY

C U R R I C U L U M

Get the right training to build secure applications.

CORE

STH.DEVELOPER
Application Security Awareness Modules

DEV522
Defending Web Applications Security Essentials
GWEB

DEV531
Defending Mobile Applications Security Essentials

DEV534
Secure DevOps: A Practical Introduction

SECURE CODING

DEV541
Secure Coding in Java/JEE
GSSP-JAVA

DEV544
Secure Coding in .NET
GSSP-NET

DEV543
Secure Coding in C/C++

SPECIALIZATION

SEC542
Web App Penetration Testing and Ethical Hacking
GWAPT

SEC642
Advanced Web App Penetration Testing and Ethical Hacking

ASSESSMENT

AppSec CyberTalent Assessment
sans.org/appsec-assessment

@sansappsec

software-security.sans.org

SANS

Questions?

ejohnson@sans.org
@emjohn20