RSAConference2019
San Francisco | March 4–8 | Moscone Center

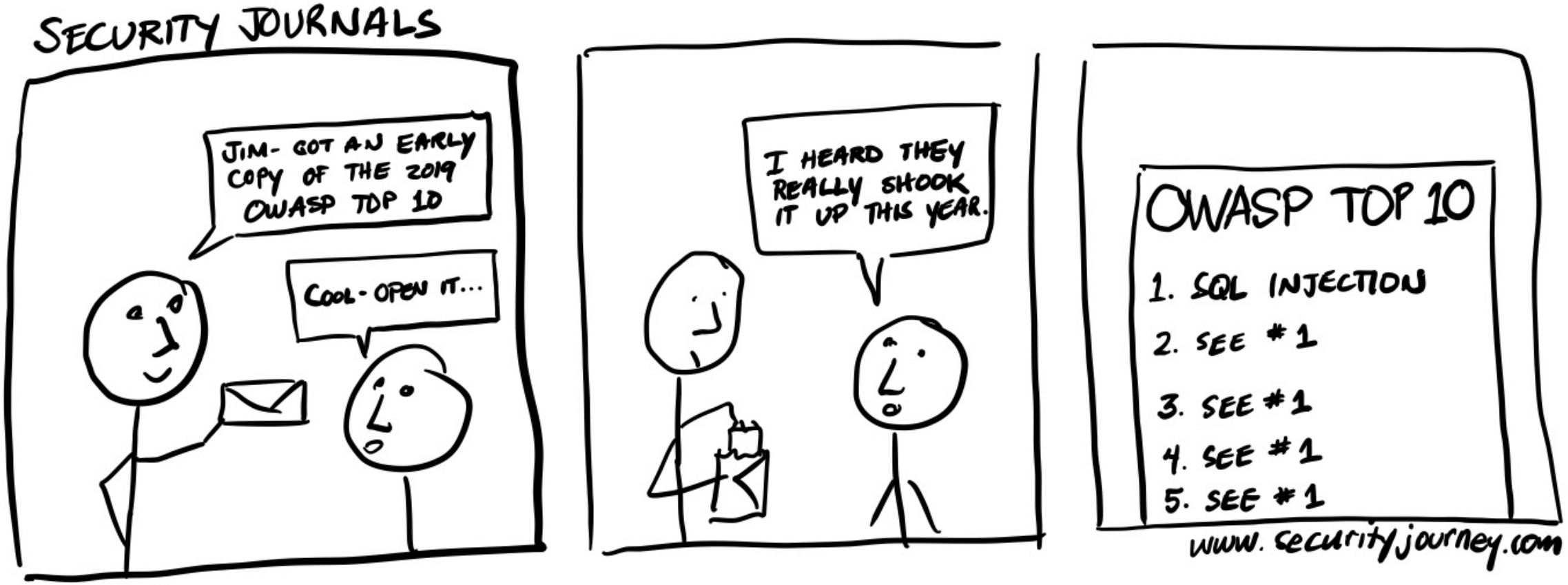
SESSION ID: HUM-R02

Cheaper by the dozen: application security on a limited budget

Christopher J. Romeo
CEO
Security Journey
@edgeroute

Agenda

1. Traditional application security programs
2. The importance of security community
3. Building a program based on OWASP
   - Awareness and education
   - Process and measurement
   - Tools
4. Final thoughts


SECURITY JOURNALS
JIM- GOT AN EARLY COPY OF THE 2019 OWASP TOP 10
COOL- OPEN IT...
I HEARD THEY REALLY SHOOK IT UP THIS YEAR.
OWASP TOP 10
1. SQL INJECTION
2. SEE #1
3. SEE #1
4. SEE #1
5. SEE #1
www.securityjourney.com

Traditional AppSec programs

PEOPLE

PROCESS

TOOLS

Goals of an AppSec Program

GOAL 1

Limit vulnerabilities in deployed code

GOAL 2

Build secure software and teach developers to build secure software

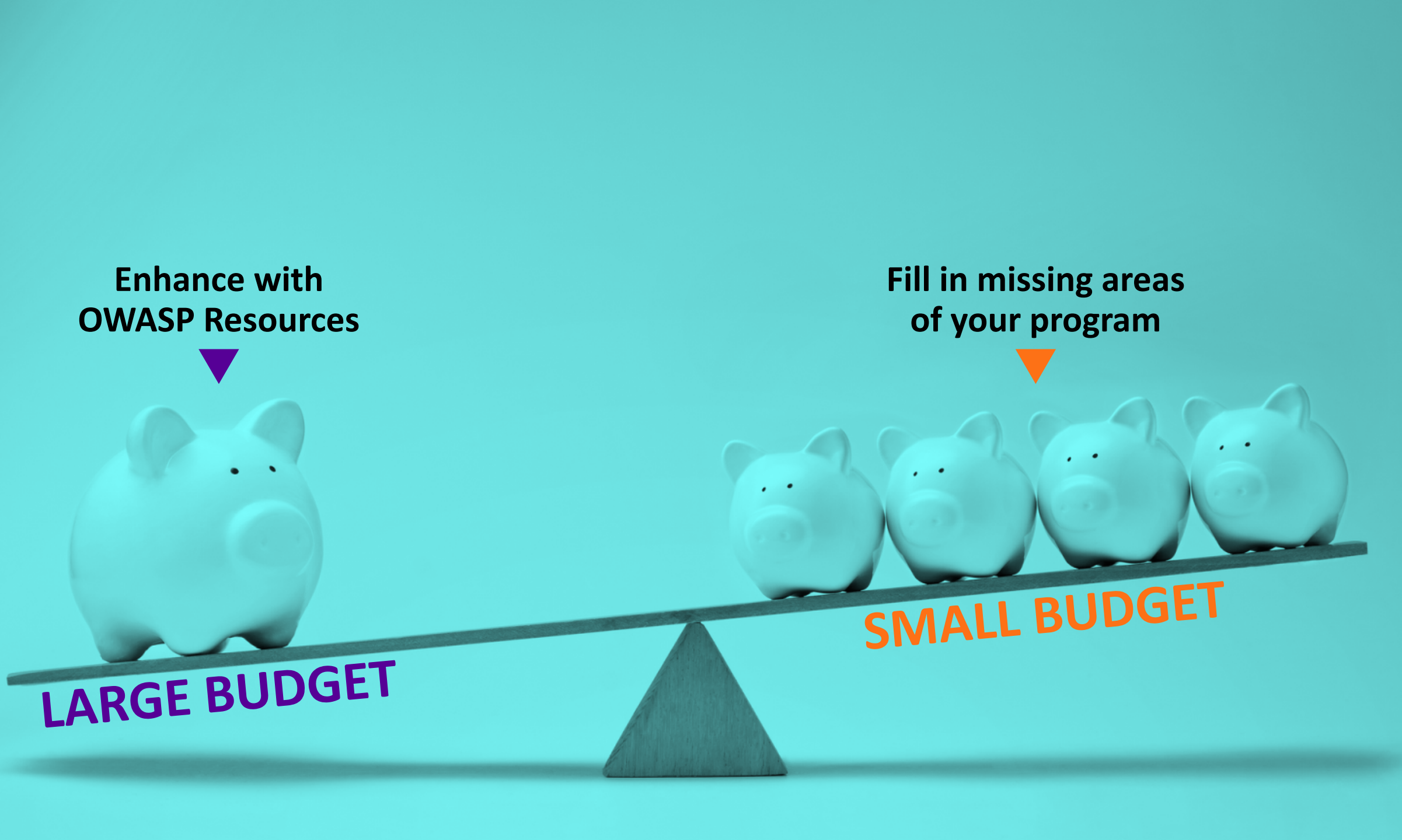
GOAL 3

Provide processes and tools for AppSec standardization

GOAL 4

Demonstrate software security maturity through metrics and assessment

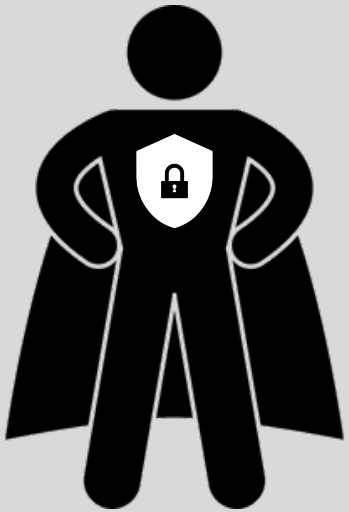
What if I had to develop an application security program with a budget of $0?


Enhance with
OWASP Resources
Fill in missing areas
of your program
LARGE BUDGET
SMALL BUDGET

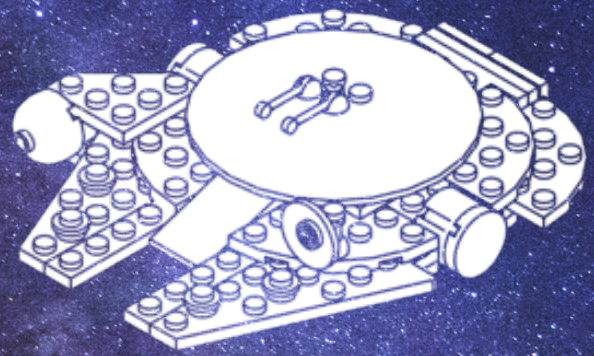
Security Champions

se · cu · ri · ty cham · pi · on [*sih · kyer · uh · tee cham · pee · uhn*], noun **1** a person passionate about security with a desire to educate those around them.

we all want to embed security champions in our companies.



OWASP



FLAGSHIP PROJECTS

13

LAB PROJECTS

35

INCUBATOR PROJECTS

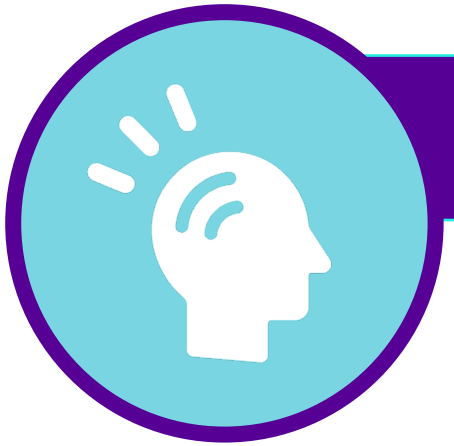
49

Scale of project risk

Rating	Explanation
0	The only way this goes away is if owasp.org disappears off the Internet
1-3	Stable project, multiple releases, high likelihood of sustainability
4-6	Newer project, fewer releases
7-9	Older project with a lack of updates within the last year
10	If I added one of these to this project, I should have my head examined

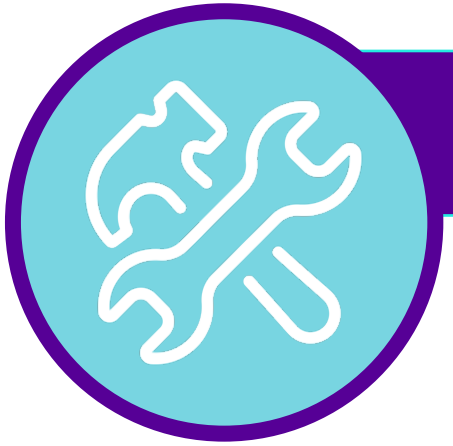

NOTICE
Use OWASP projects with caution. There is no guarantee that a project will ever be updated again.

The categories



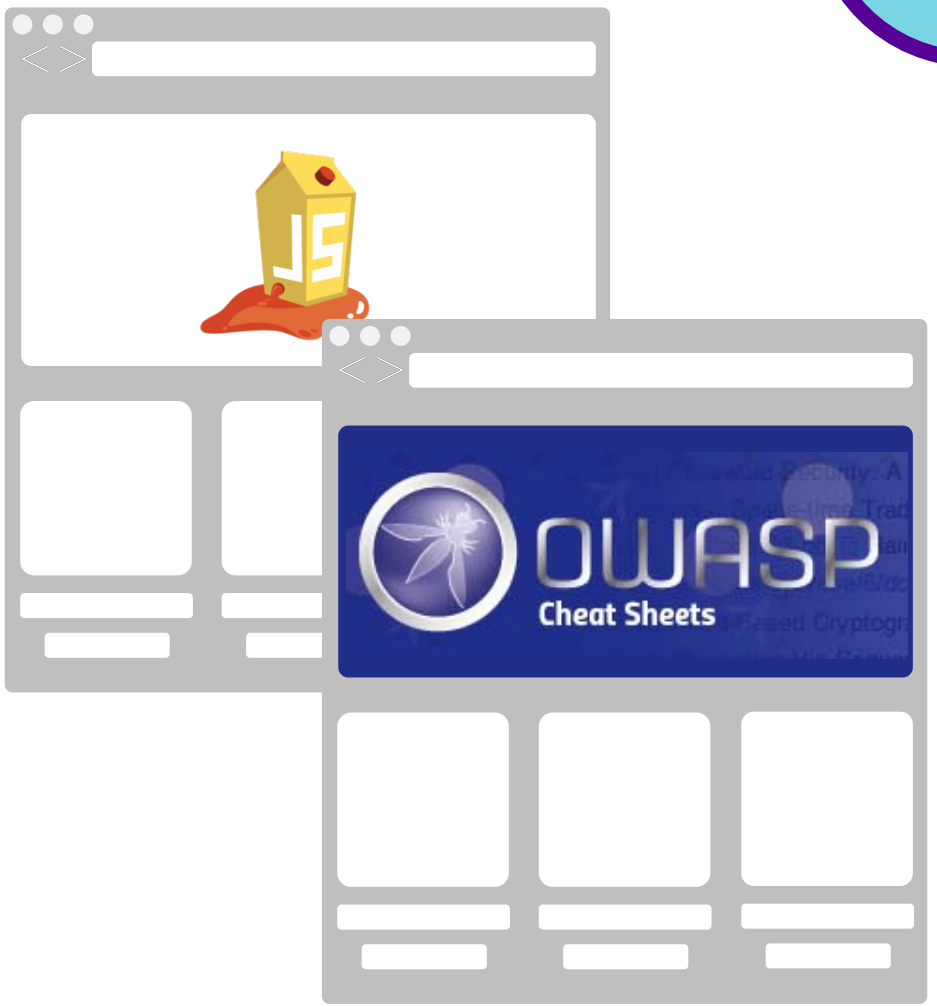
Awareness, knowledge, and education

Process and measurement



Tools

Awareness, knowledge and education



Project Risk 0

A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

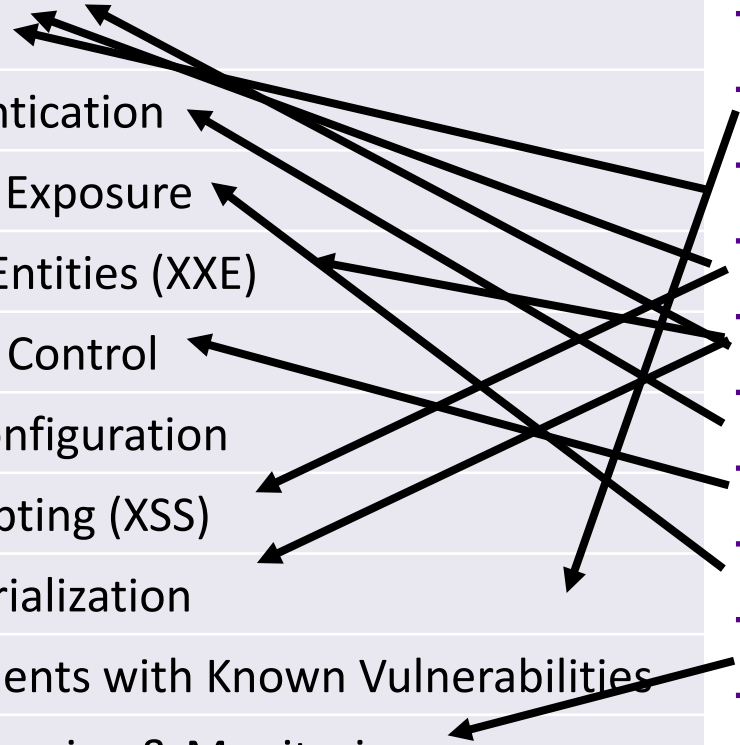
Project Risk 2

C1 Define Security Requirements	C2 Leverage Security Frameworks and Libraries	C3 Secure Database Access	C4 Encode and Escape Data
C5 Validate All Imputs	C6 Implement Digital Identity	C7 Enforce Access Control	C8 Protect Data Everywhere
	C9 Implement Security Logging and Monitoring	C10 Handle All Errors and Exceptions	

https://www.owasp.org/index.php/OWASP_Proactive_Controls

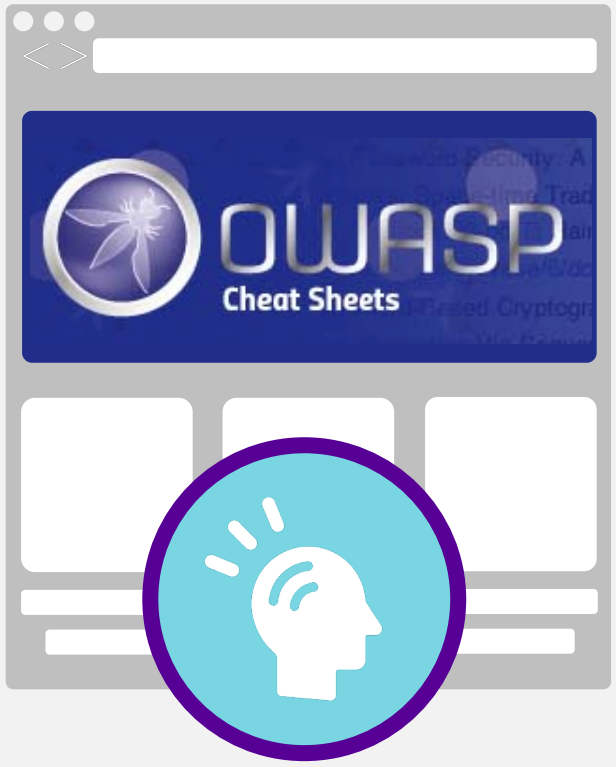
The intermingling

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring



OWASP Pro Active Controls

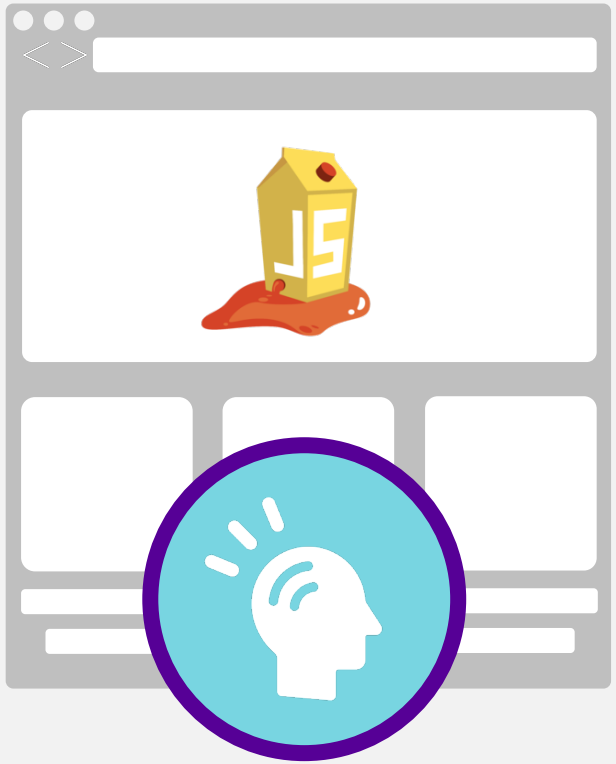
C1 Define Security Requirements
C2 Leverage Security Frameworks and Libraries
C3 Secure Database Access
C4 Encode and Escape Data
C5 Validate All Imputs
C6 Implement Digital Identity
C7 Enforce Access Control
C8 Protect Data Everywhere
C9 Implement Security Logging and Monitoring
C10 Handle All Errors and Exceptions



Project Risk 2

V · T · E	Cheat Sheets [Collapse]
Developer / Builder	3rd Party Javascript Management · Access Control · AJAX Security Cheat Sheet · Authentication (ES) · Bean Validation Cheat Sheet · Choosing and Using Security Questions · Clickjacking Defense · Credential Stuffing Prevention Cheat Sheet · Cross-Site Request Forgery (CSRF) Prevention · Cryptographic Storage · C-Based Toolchain Hardening · Deserialization · DOM based XSS Prevention · Forgot Password · HTML5 Security · HTTP Strict Transport Security · Injection Prevention Cheat Sheet · Injection Prevention Cheat Sheet in Java · JSON Web Token (JWT) Cheat Sheet for Java · Input Validation · Insecure Direct Object Reference Prevention · JAAS · Key Management · LDAP Injection Prevention · Logging · Mass Assignment Cheat Sheet · .NET Security · OS Command Injection Defense Cheat Sheet · OWASP Top Ten · Password Storage · Pinning · Query Parameterization · REST Security · Ruby on Rails · Session Management · SAML Security · SQL Injection Prevention · Transaction Authorization · Transport Layer Protection · Unvalidated Redirects and Forwards · User Privacy Protection · Web Service Security · XSS (Cross Site Scripting) Prevention · XML External Entity (XXE) Prevention Cheat Sheet
Assessment / Breaker	Attack Surface Analysis · REST Assessment · Web Application Security Testing · XML Security Cheat Sheet · XSS Filter Evasion
Mobile	Android Testing · IOS Developer · Mobile Jailbreaking
OpSec / Defender	Virtual Patching · Vulnerability Disclosure
Draft and Beta	Application Security Architecture · Business Logic Security · Content Security Policy · Denial of Service Cheat Sheet · Grails Secure Code Review · IOS Application Security Testing · PHP Security · Regular Expression Security Cheatsheet · Secure Coding · Secure SDLC · Threat Modeling

https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series



Project Risk 3

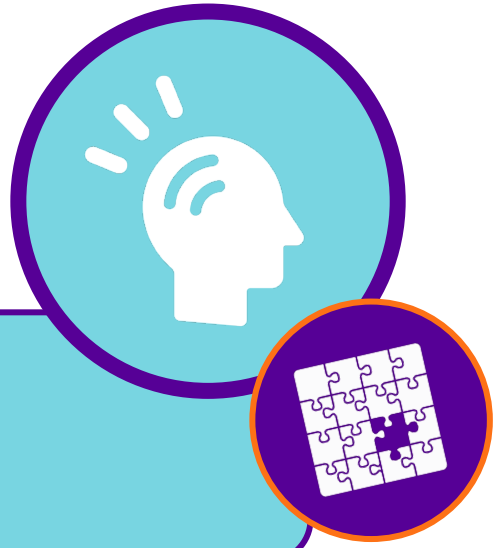
JavaScript-based

Intentionally insecure web app

Encompasses the entire OWASP Top Ten and other severe security flaws

https://www.owasp.org/index.php/OWASP_Juice_Shop_Project

Missing pieces in awareness, knowledge and education



Delivery of awareness and education

Administration of the training platforms

Awareness and education: impact and headcount

Awareness

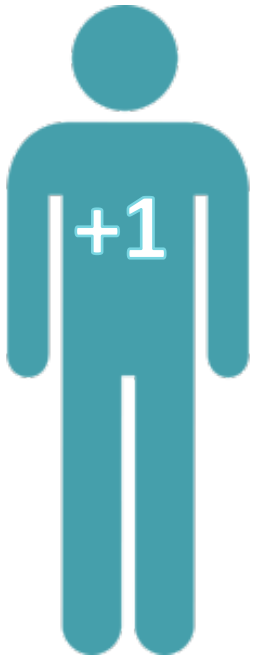
Foundational understanding of the most important concepts in AppSec

Knowledge

A concise reference for solving the most difficult AppSec problems

Hands-on training

Assimilation of key concepts through activities that lock in knowledge and make it practical



Awareness and education: getting started

Awareness

Lunch and learn sessions to teach the basics of all awareness documents

Knowledge

Teach developers about available cheat sheets

Host an internal copy of the cheat sheets

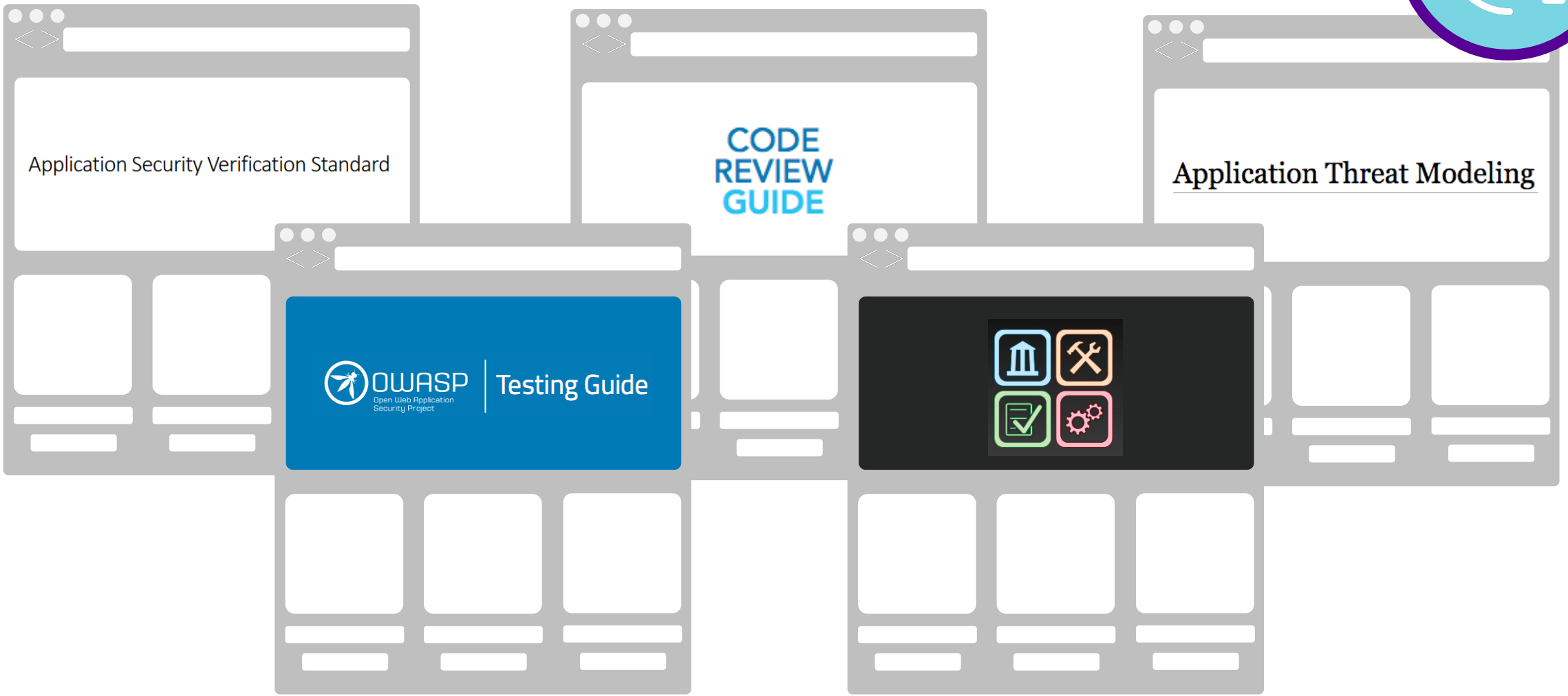
Lead a training session covering the three most crucial cheat sheets for your organization

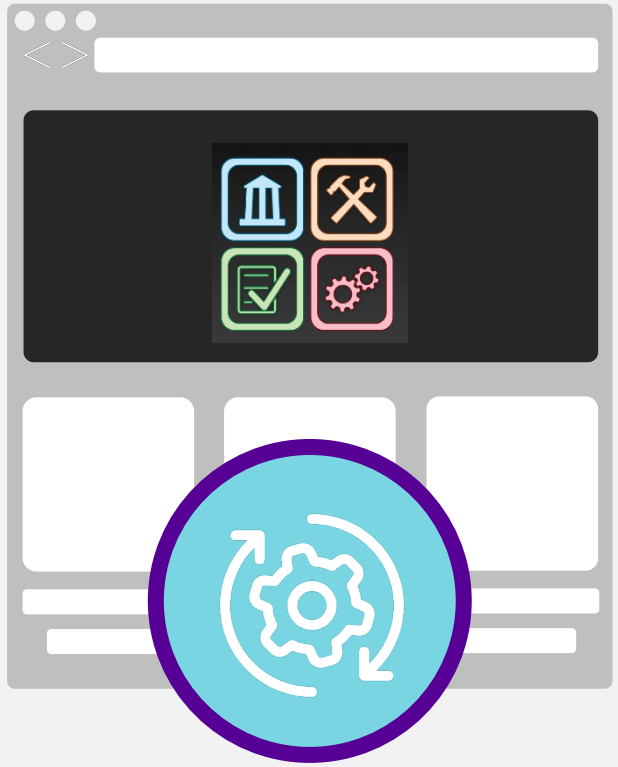
Hands-on Training

Build an environment that hosts JuiceShop

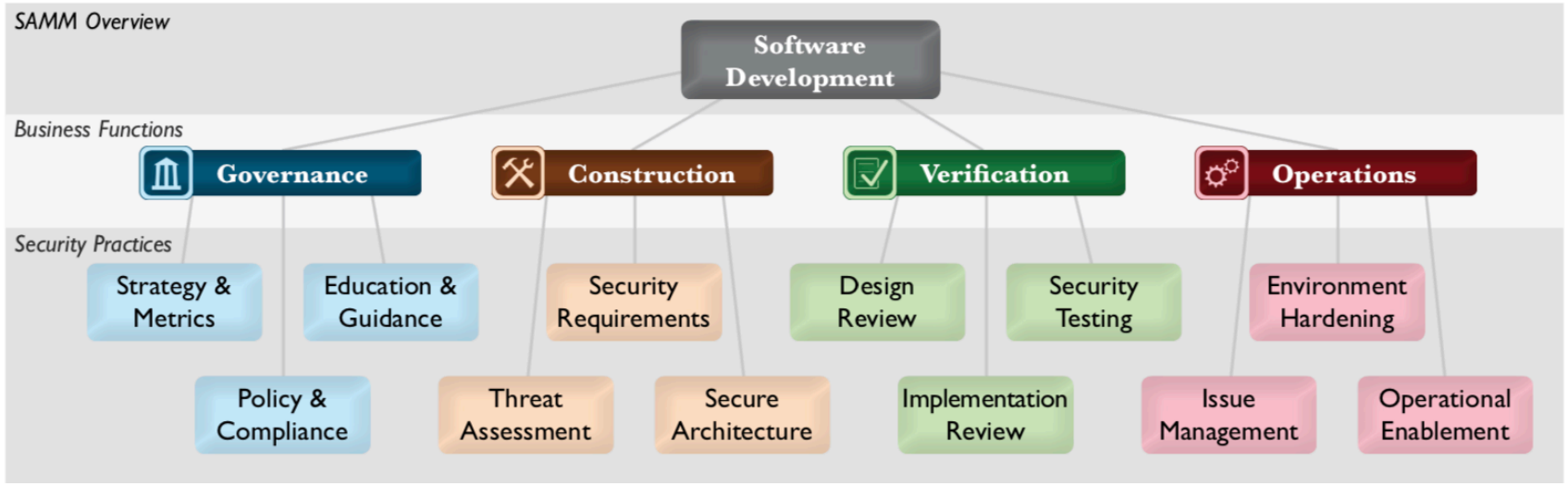
Schedule a hack-a-thon where teams gather together and work on JuiceShop in teams and learn from each other

Process and Measurement





Project Risk 1



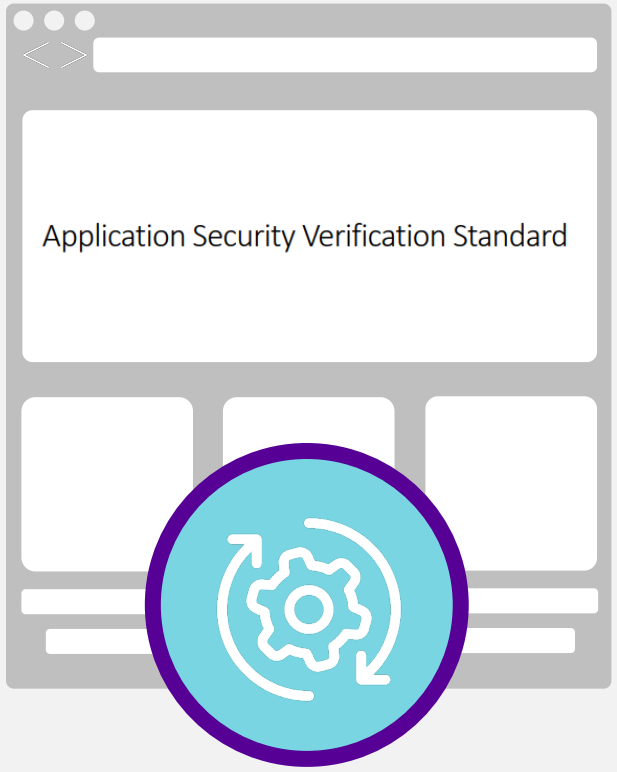
0 Implicit starting point representing the activities in the practice being unfulfilled

1 Initial understanding and adhoc provision of security practice

2 Increase efficiency and/or effectiveness of the security practice

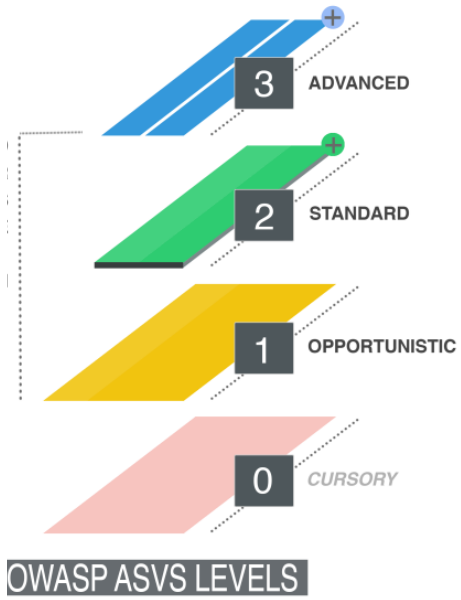
3 Comprehensive mastery of the security practice at scale

https://www.owasp.org/index.php/OWASP_SAMM_Project



Project Risk 1

Requirement	
V1. Architecture, design and threat modelling	V11. HTTP security configuration
V2. Authentication	V13. Malicious controls
V3. Session management	V15. Business logic
V4. Access control	V16. File and resources
V5. Malicious input handling	V17. Mobile
V7. Cryptography at rest	V18. Web services
V8. Error handling and logging	V19. Configuration
V9. Data protection	V11. HTTP security configuration
V10. Communications	

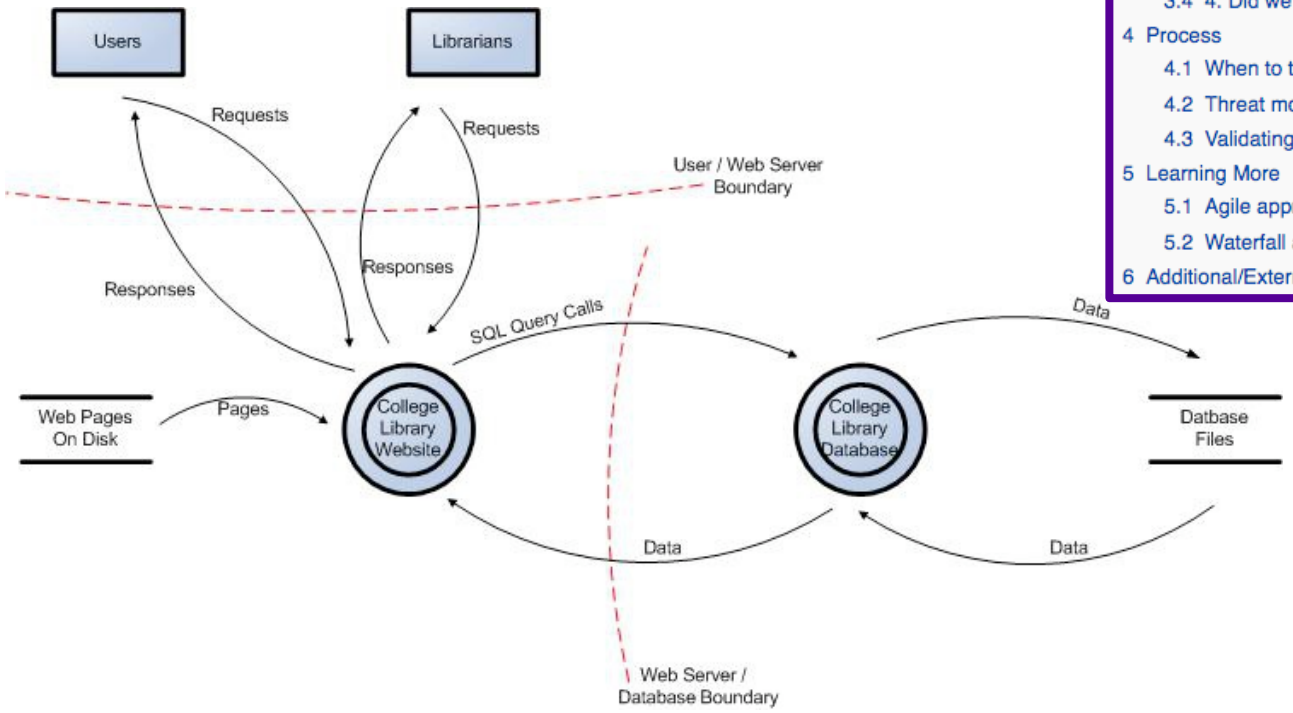


https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

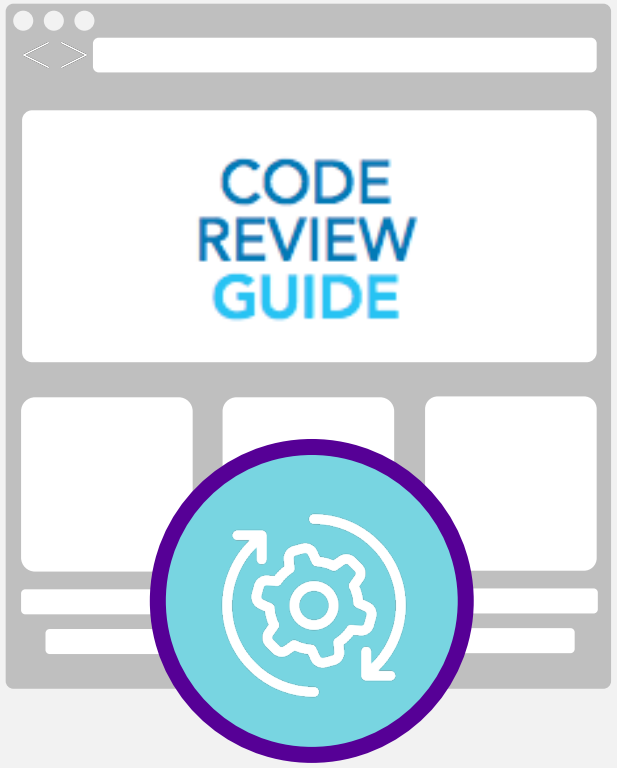
Application Threat Modeling

Project Risk 5

1 What
2 Why
3 4 Questions
  3.1 1. What are we building?
  3.2 2. What can go wrong?
  3.3 3. What are we going to do about that?
  3.4 4. Did we do a good enough job?
4 Process
  4.1 When to threat model
  4.2 Threat modelling: engagement versus review
  4.3 Validating assumptions
5 Learning More
  5.1 Agile approaches
  5.2 Waterfall approaches
6 Additional/External references



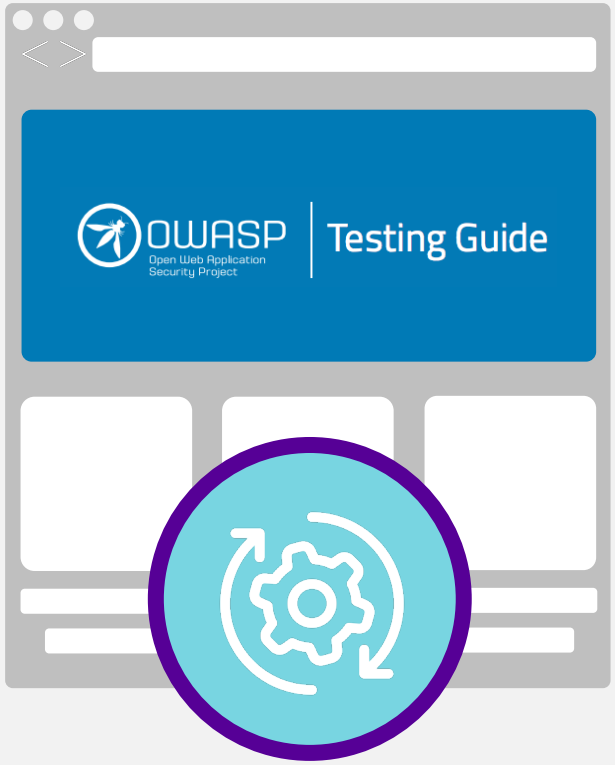
https://www.owasp.org/index.php/Application_Threat_Modeling



Project Risk 4

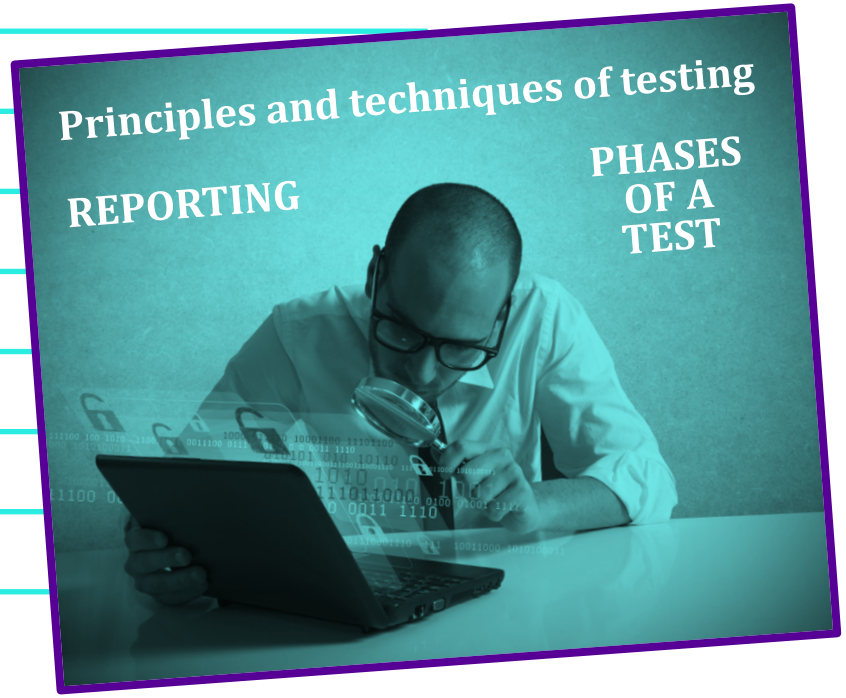
- Secure code review methodology
- Technical reference for secure code review: OWASP Top 10
- HTML5
- Same origin policy
- Reviewing logging code
- Error handling
- Buffer overruns
- Client side JavaScript
- Code review do's and don'ts
- Code review checklist
- Code crawling

https://www.owasp.org/index.php/Category:OWASP_Code_Review_Project



Project Risk 1

- Information gathering
- Configuration and deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak crypto
- Business logic testing
- Client side testing



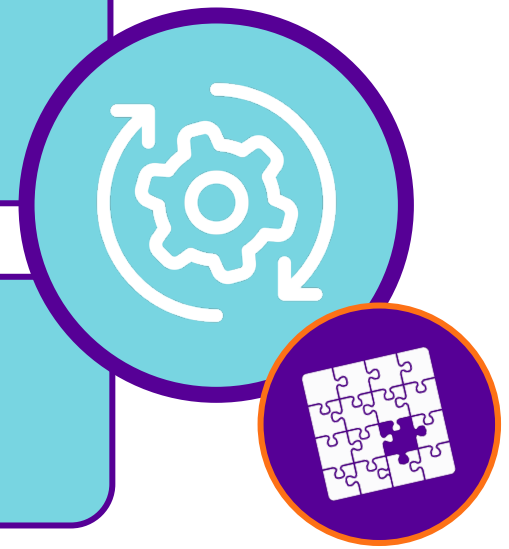
https://www.owasp.org/index.php/OWASP_Testing_Project

Missing pieces in process and measurement

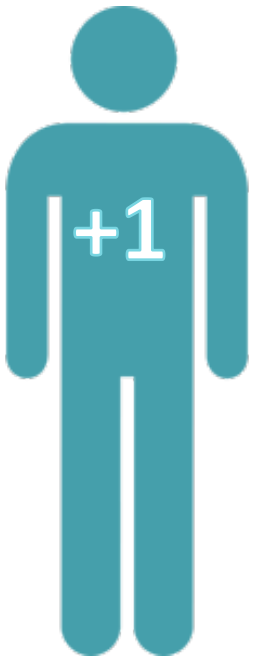
End-to end SDL or Secure SDLC

Program metrics

Deployment advice/experience on how to be successful



Process and measurement: impact and headcount



Process

ASVS provides important requirements

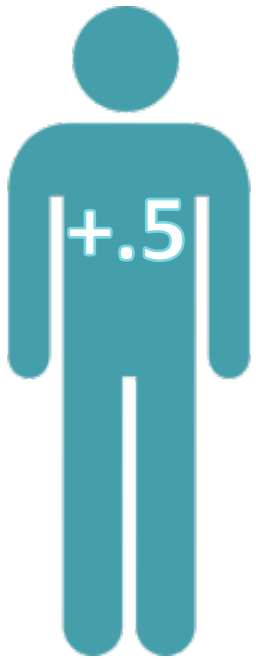
App threat modeling defines the process with examples

Code review guide describes how to perform a code review and what to look for

Testing guide provides how to test and a knowledge base of how to exploit vulnerabilities

Measurement

A roadmap to where you are today, and a plan for where you want to go with your AppSec program



Process and measurement: getting started

Process

Choose one of the process areas to start with (threat modeling) and build out this activity as your first

Early wins are key!

Measurement

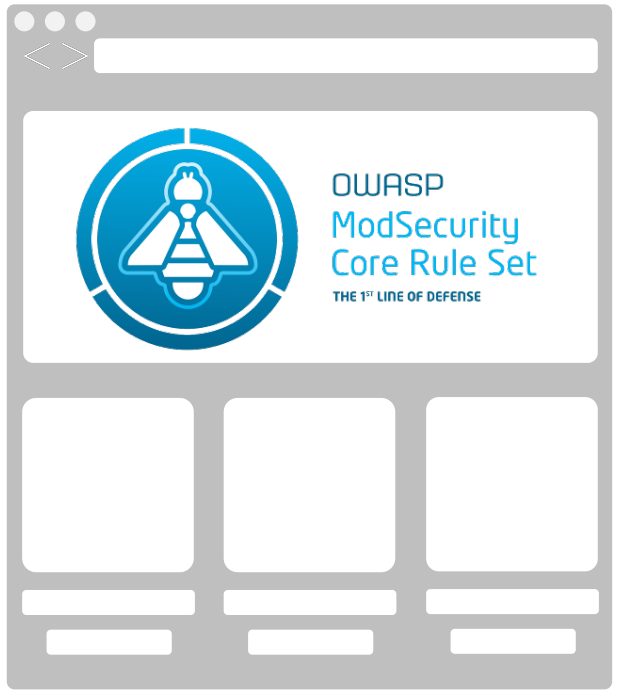
Perform an early assessment to determine where you are

Map out a future plan for where you want to get to

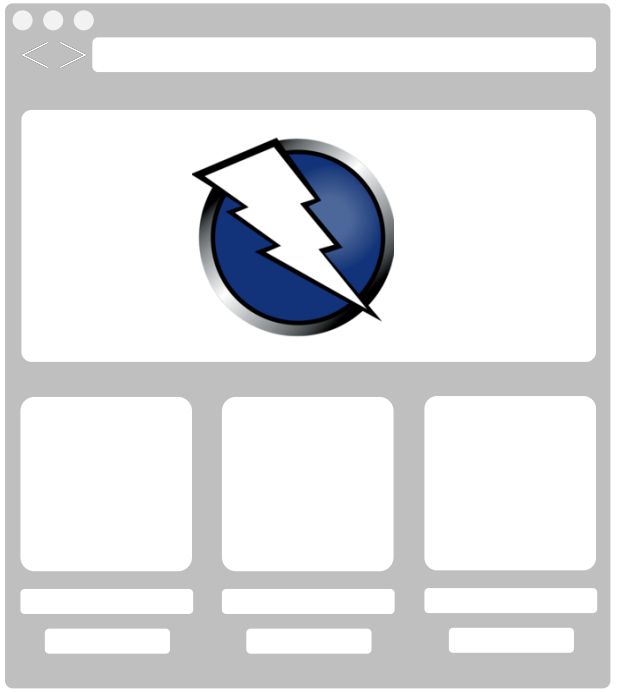
Share these assessments with Executives and Security Champions (and anyone else that will listen)

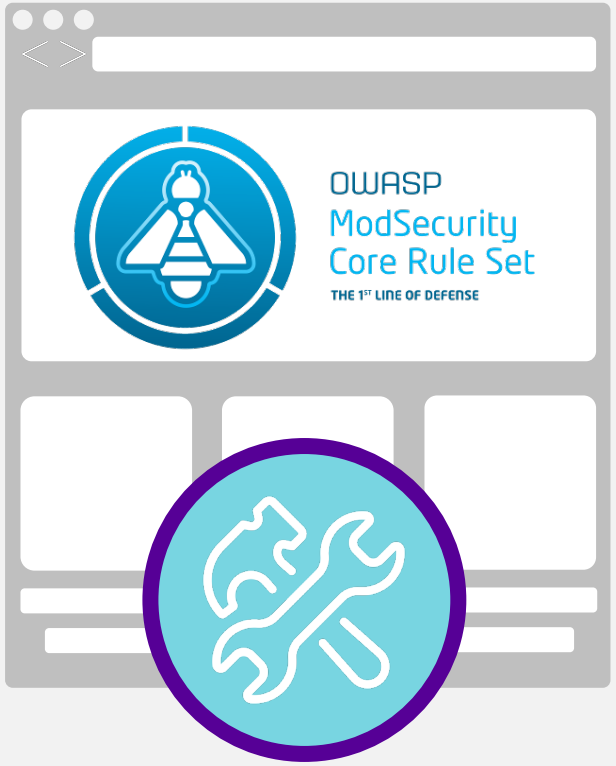
Advocate for Executive support on your plan to build a stronger AppSec program

Tools

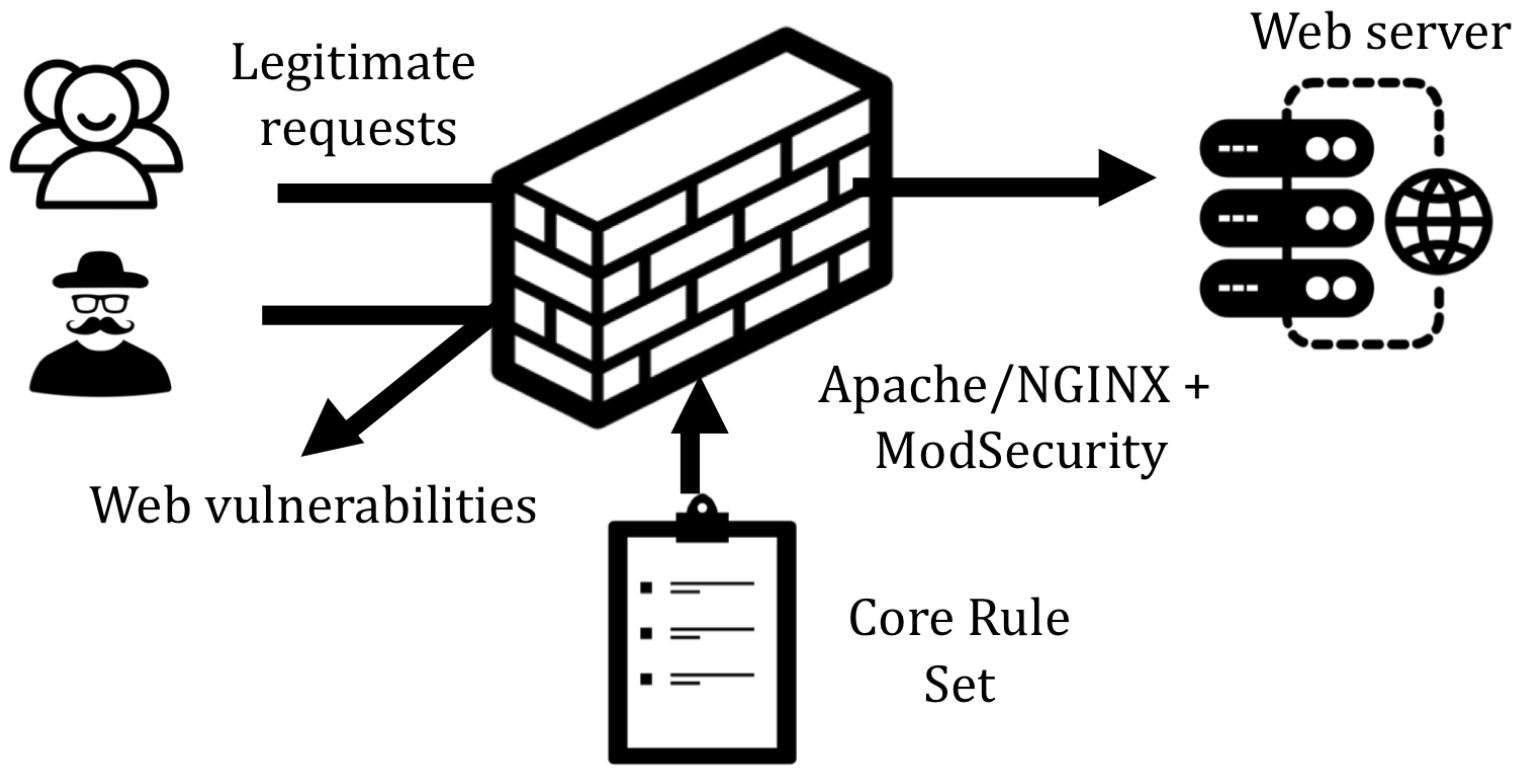

OWASP
ModSecurity
Core Rule Set
THE 1ST LINE OF DEFENSE


DEPENDENCY-CHECK

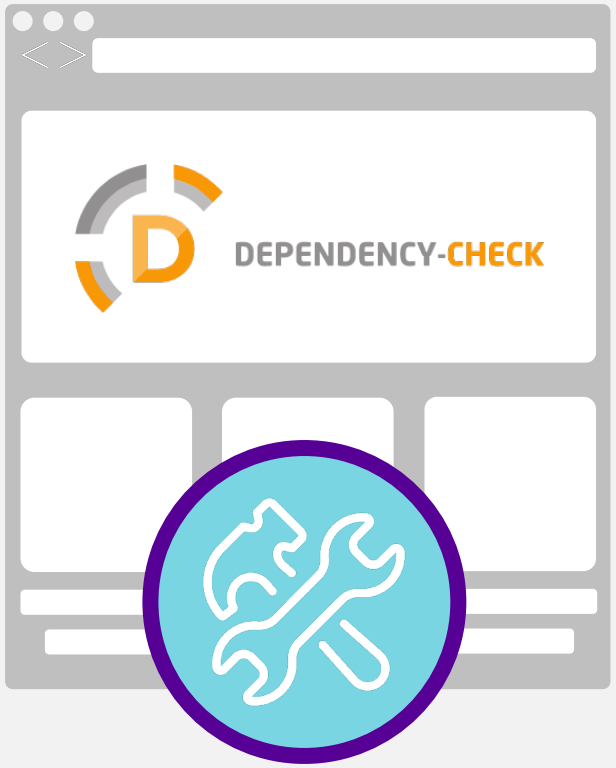




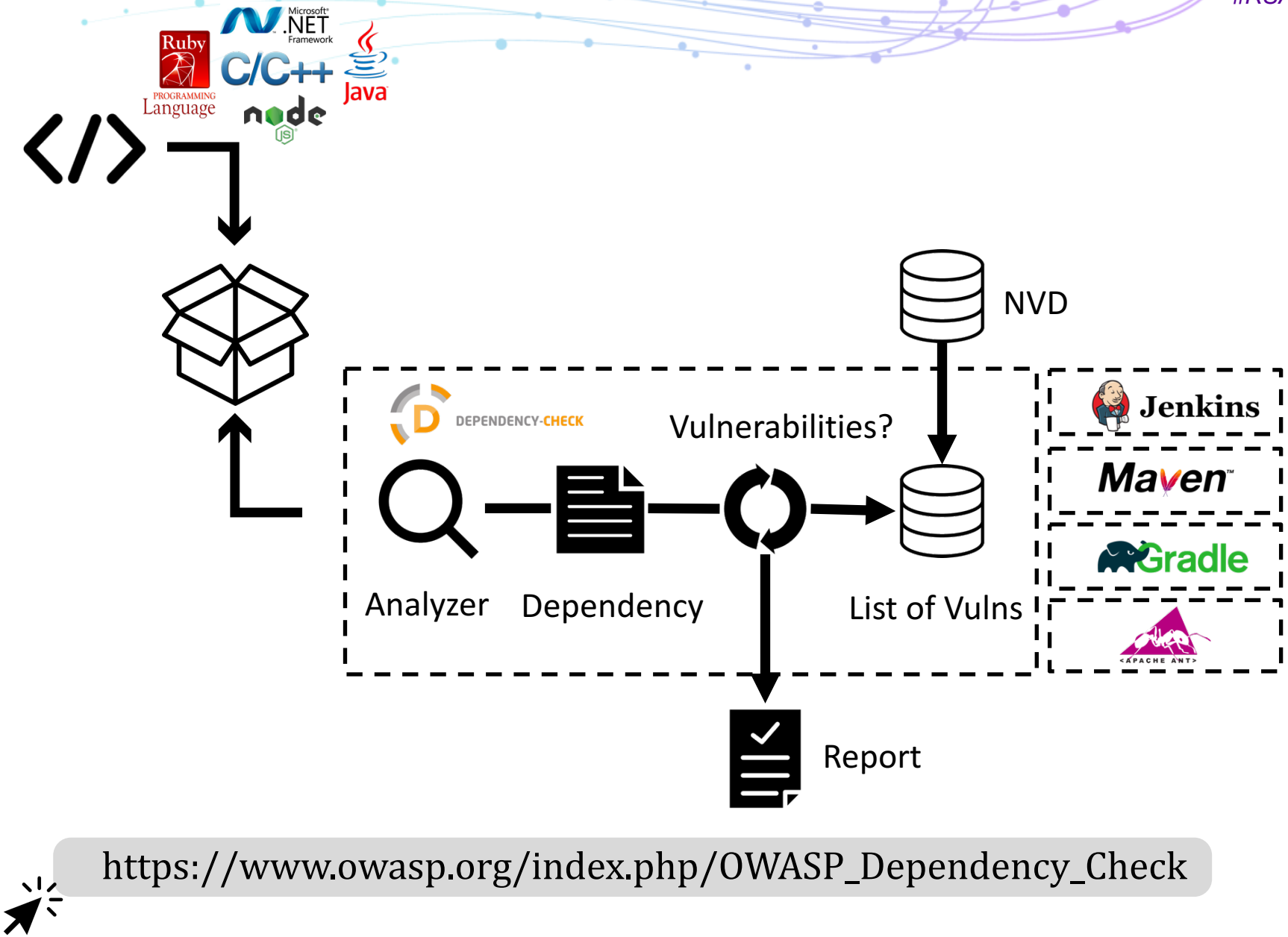
Project Risk 1



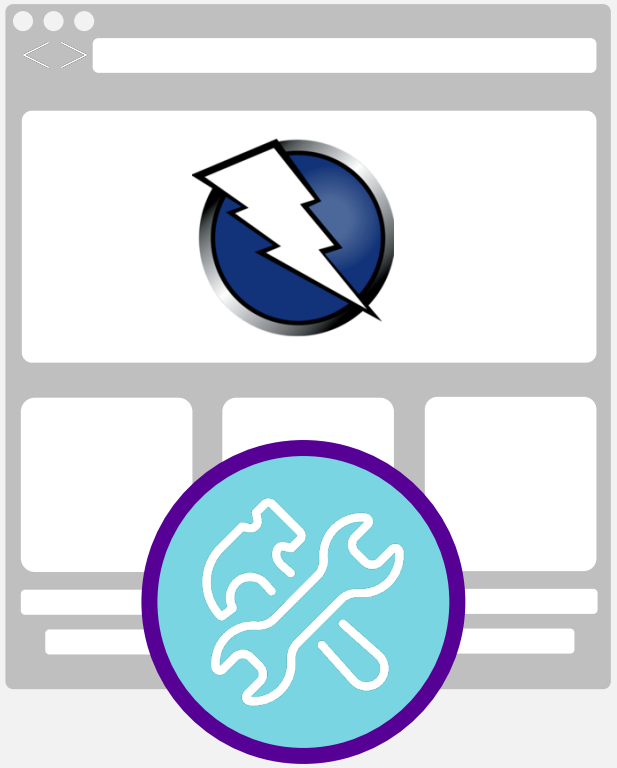
https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project



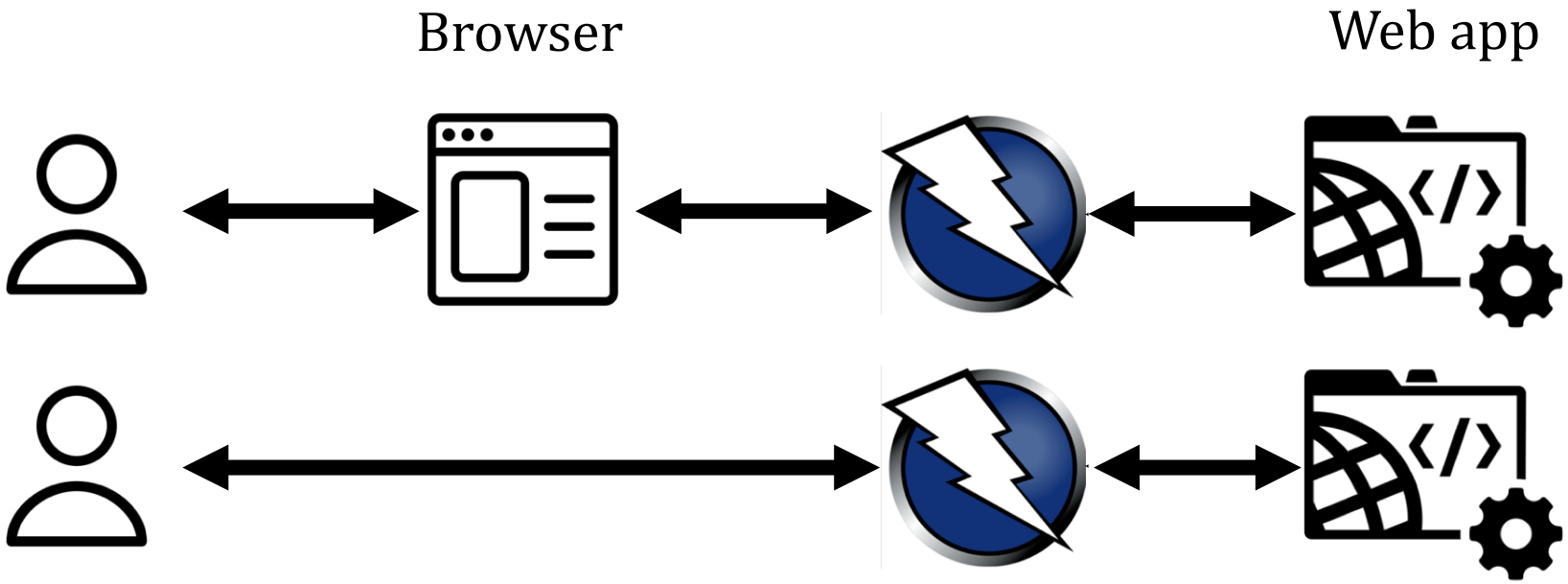
Project Risk 3



https://www.owasp.org/index.php/OWASP_Dependency_Check



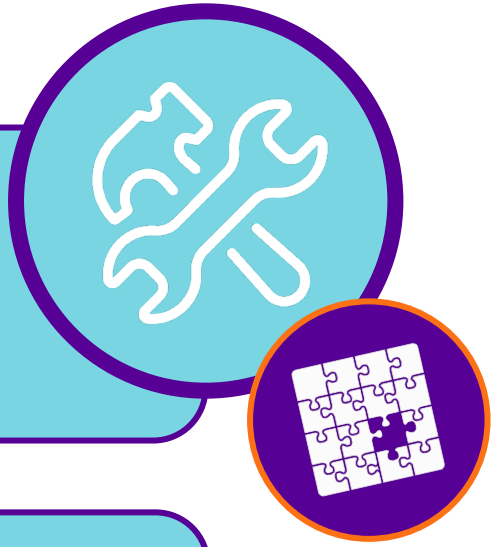
Project Risk 2



https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

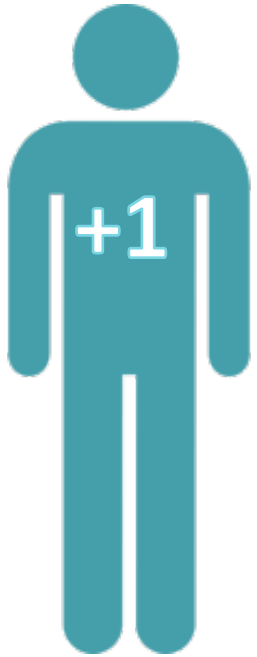
Missing pieces in tools

No options for SAST or IAST



A dashboard to track everything (requirements management, activities, releases, metrics)

Tools: impact and headcount

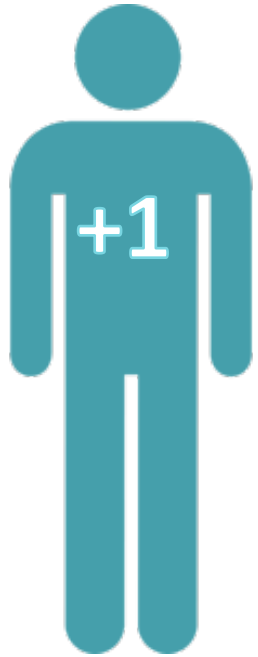


Infrastructure

CRS provides a true WAF solution

Dependency check identifies vulnerable 3rd party software

ZAP provides DAST, and plugs in to any dev methodology



Tools: getting started

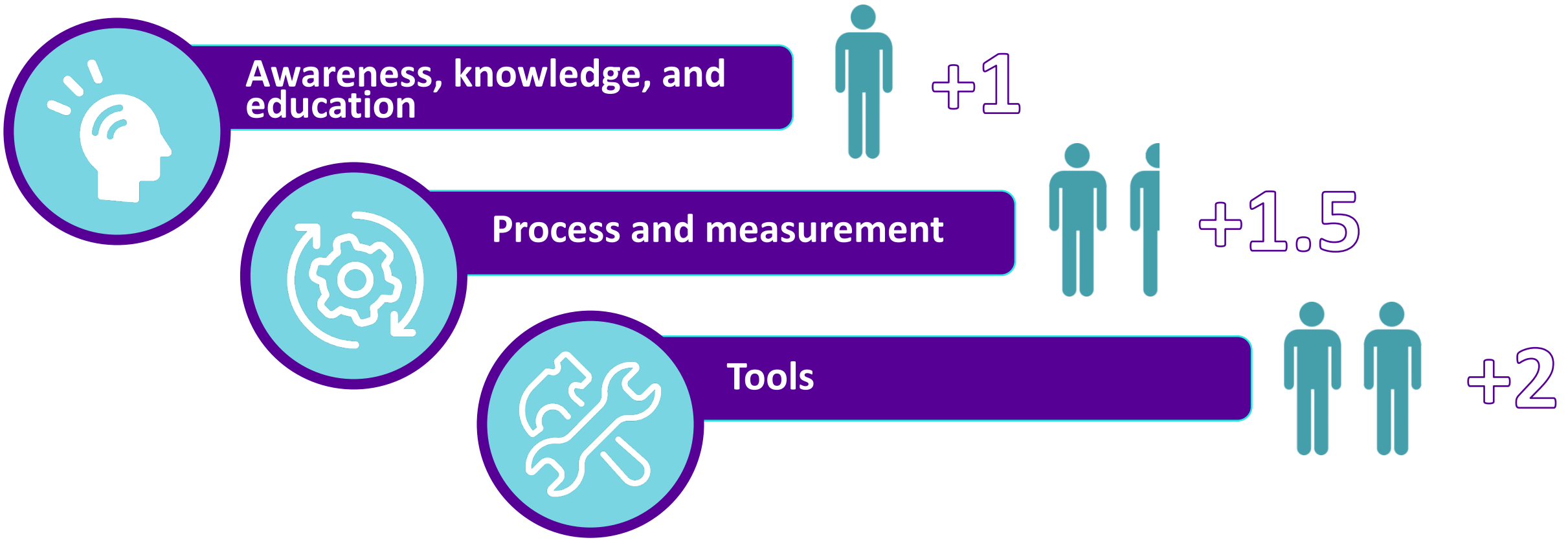
Infrastructure

Add Dependency Check to your build pipeline tomorrow

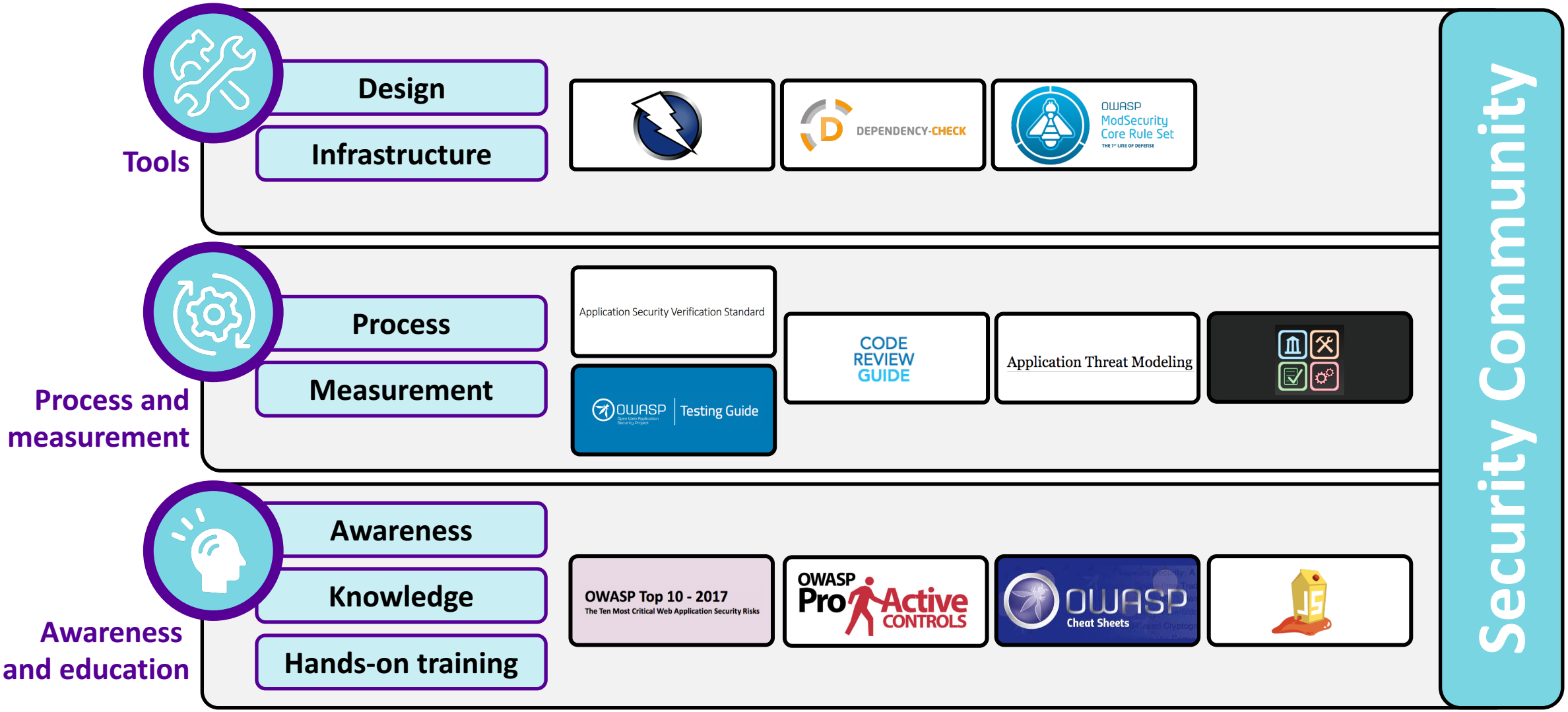
Teach ZAP to Security Champions and interested testers

Work with your infra owner to deploy a test of ModSecurity + CRS

Headcount summary



The dozen OWASP projects as an AppSec program



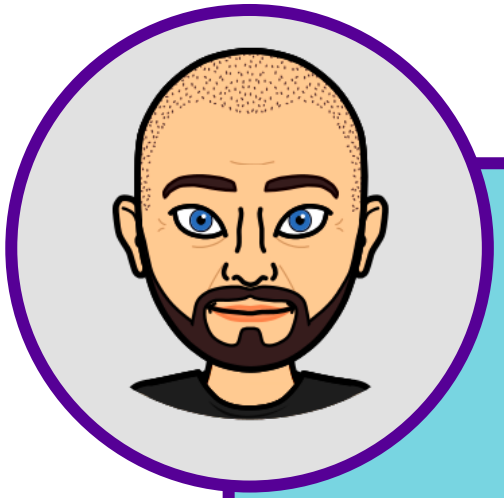
Apply What You Have Learned Today

- Next week you should:
  - Assess a high-level current state of your application security program and determine if you have visible gaps
- In the first three months following this presentation you should:
  - Perform a deeper assessment using OpenSAMM
  - Choose one of the dozen to implement
- Within six months you should:
  - Measure the impact of your first project implementation
  - Plan and execute on one or two additional pieces, resources permitting

Final thoughts for an AppSec program on the cheap

1. Use Open SAMM to assess current program and future goals.
2. There is no OWASP SDL; build/tailor required.
3. Start small; choose one item for awareness and education to launch your program.
4. Build security community early; it is the support structure.
5. Evaluate available projects in each category and build a 1-2 year plan to roll each effort out.
6. While OWASP is free, head count is not; plan for head count to support your “free” program.

Q+A and Thank you!



CHRIS ROMEO

CEO / Co-Founder

chris_romeo@securityjourney.com

www.securityjourney.com

@edgeroute, @SecurityJourney, @AppSecPodcast