OWASP
AppSec Europe
London 2nd-6th July 2018

Building an AppSec Program with a Budget of $0: Beyond the OWASP Top 10

Chris Romeo

About Chris Romeo

- **CEO / Co-Founder @ Security Journey**
- **20 years in the security world, CISSP, CSSLP**
- **Co-host of the Application Security Podcast**
- **Co-Lead of the OWASP Triangle Chapter**

@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

Traditional AppSec programs

People

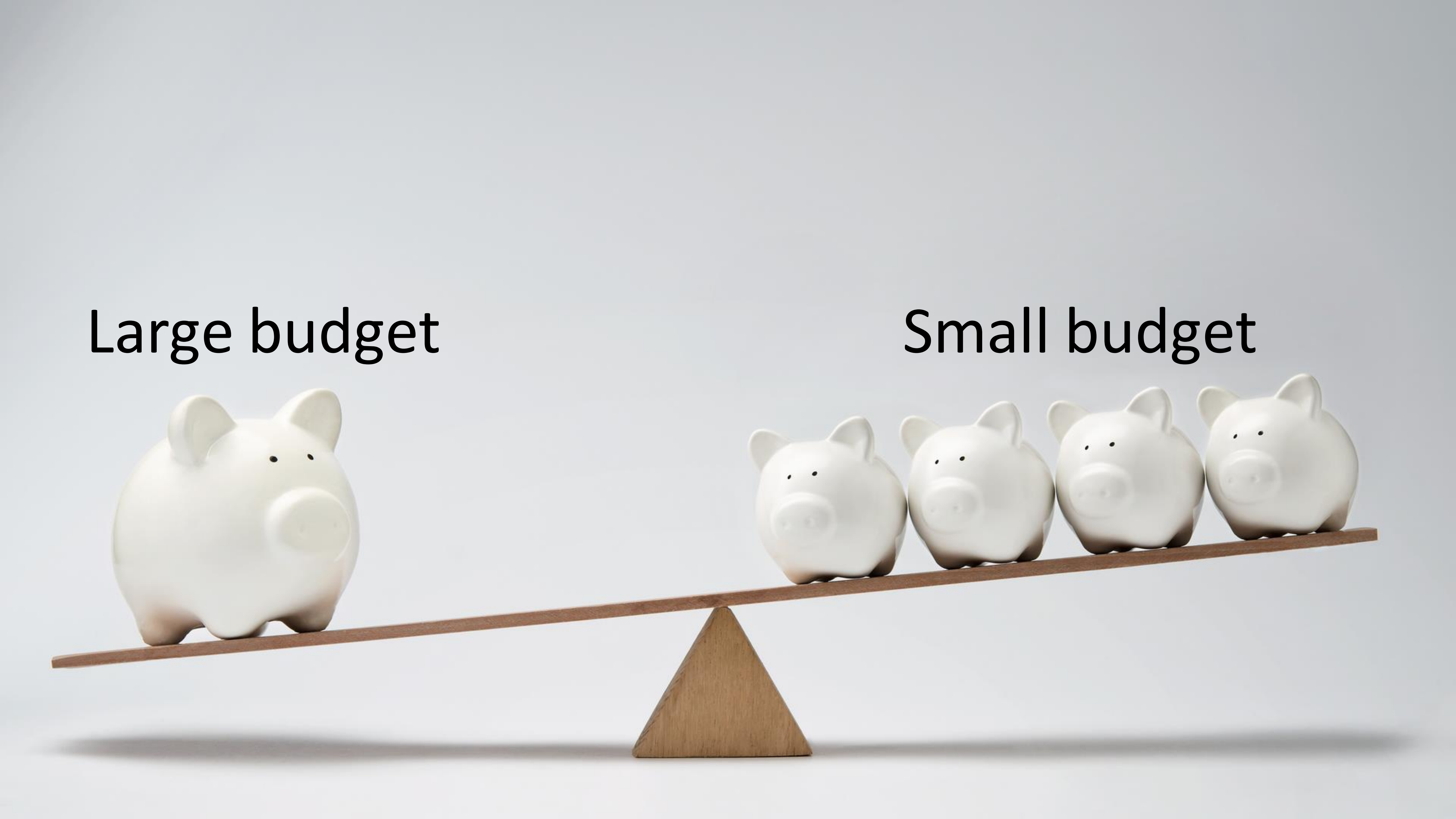
Process

Tools

Goals of an AppSec Program

1. **Limit vulnerabilities in deployed code**
2. **Build secure software and teach developers to build secure software**
3. **Provide processes and tools for AppSec standardization**
4. **Demonstrate software security maturity through metrics and assessment**

NOTE
NO BUDGET



Large budget

Small budget

Security Champions

Goal: Educate about product security and embed expertise within every product team.

Flagship
Projects: 13

OWASP

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

OWASP Top 10 - 2017
The Ten Most Critical Web Application Security Risks

OWASP
Automated Threat Handbook
Web Applications

OWASP
Cheat Sheets

WEBGOAT

DevSlop

OWASP
ProActive
CONTROLS

Security Knowledge Framework

Awareness

Risk of using project 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

OWASP Top 10 - 2017

The Ten Most Critical Web Application Security Risks

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

Awareness

Risk of using project 2

OWASP ProActive 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

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

Awareness

Risk of using project 3

OWASP Automated Threat Handbook Web Applications

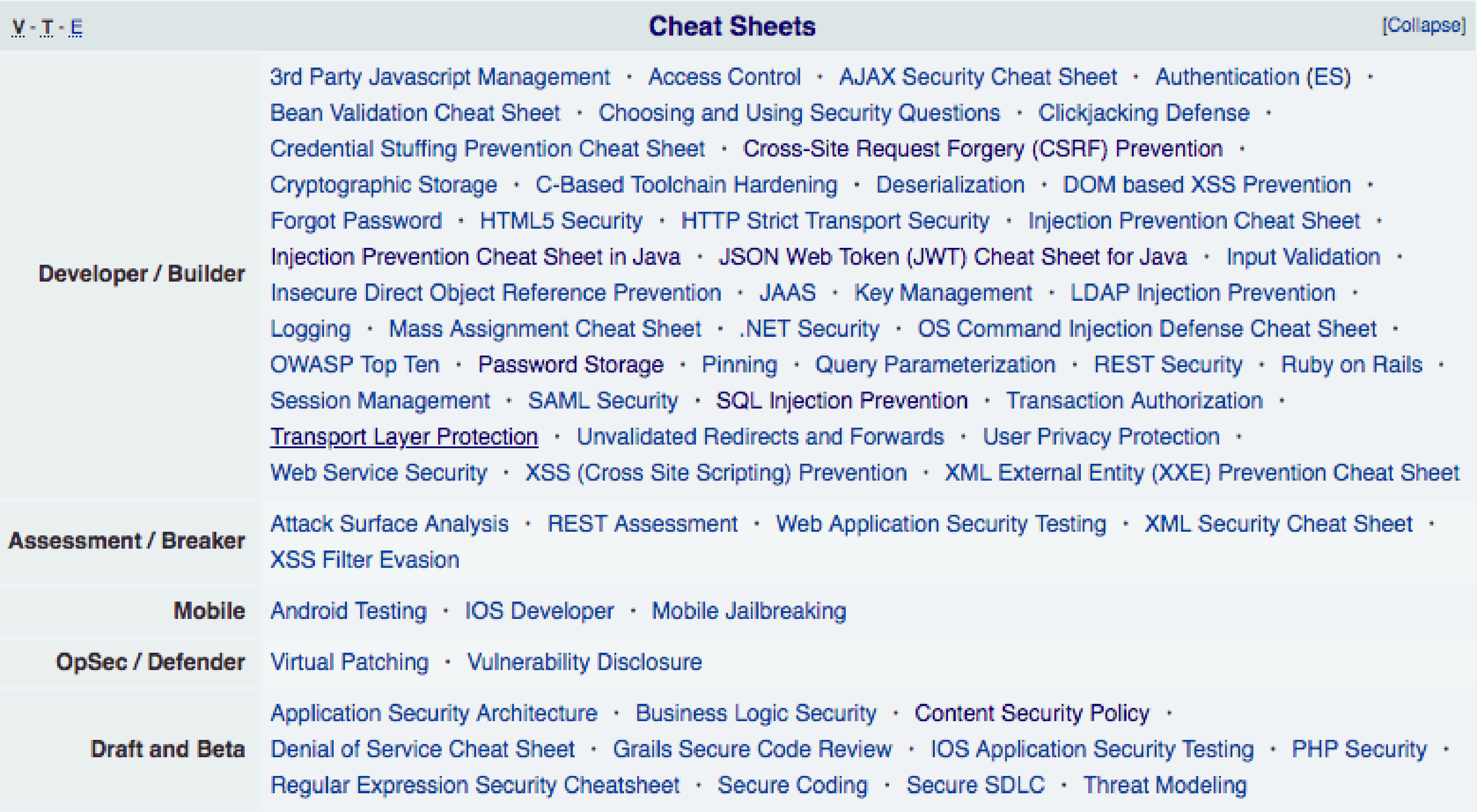
Account Aggregation	Account Creation	Ad Fraud	CAPTCHA defeat	Carding	Card Cracking	Cashing Out
Credential Cracking	Credential Stuffing	Denial of Inventory	Denial of Service	Expediting	Fingerprinting	Footprinting
Scalping	Scraping	Skewing	Sniping	Spamming	Token Cracking	Vulnerability Scanning

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

Knowledge

Risk of using project 2

OWASP
Cheat Sheets



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

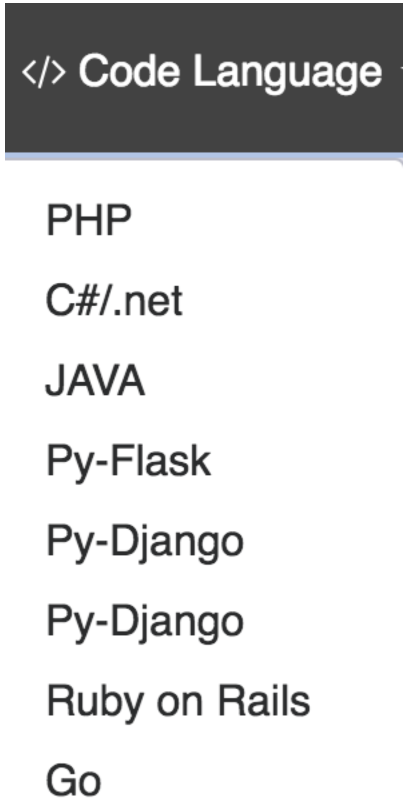
https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series

Knowledge

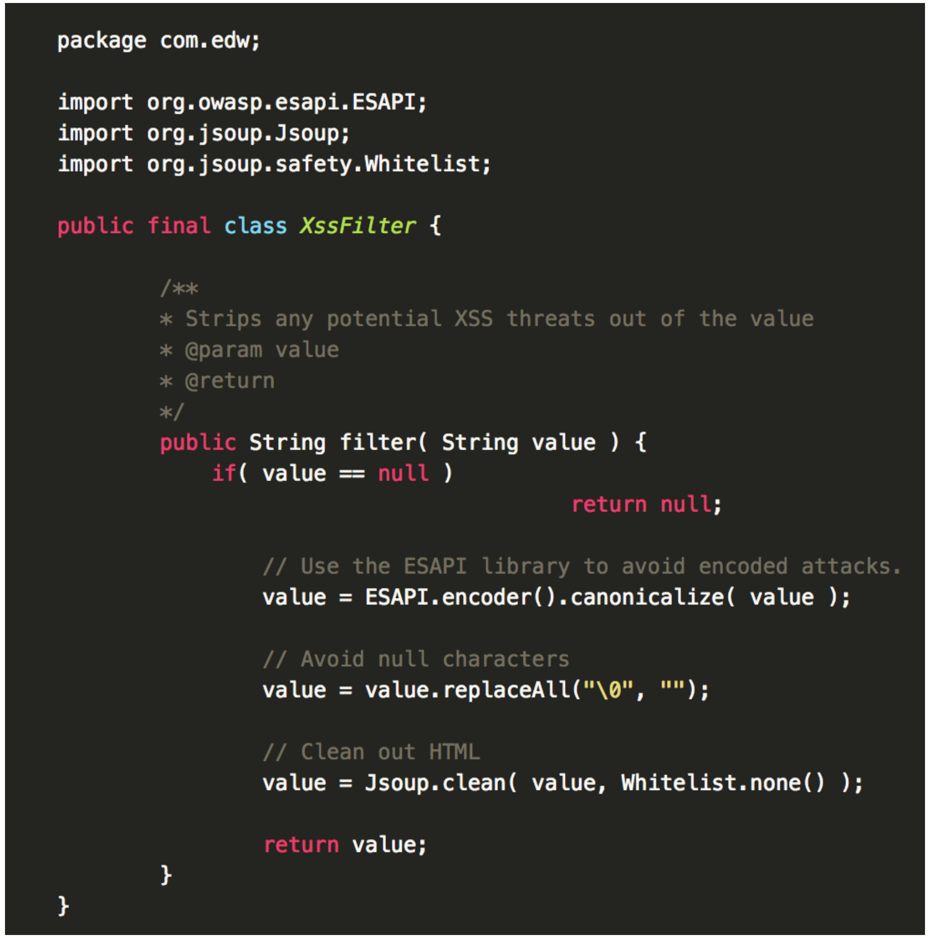
Risk of using project 4

Security Knowledge Framework

- Security Requirements OWASP ASVS for development and for third party vendor applications
- Security knowledge reference (Code examples/ Knowledge Base items)


</> Code Language

PHP
C#/.net
JAVA
Py-Flask
Py-Django
Py-Django
Ruby on Rails
Go


```java
package com.edw;

import org.owasp.esapi.ESAPI;
import org.jsoup.Jsoup;
import org.jsoup.safety.Whitelist;

public final class XssFilter {

        /**
        * Strips any potential XSS threats out of the value
        * @param value
        * @return
        */
        public String filter( String value ) {
            if( value == null )
                                        return null;

                // Use the ESAPI library to avoid encoded attacks.
                value = ESAPI.encoder().canonicalize( value );

                // Avoid null characters
                value = value.replaceAll("\0", "");

                // Clean out HTML
                value = Jsoup.clean( value, Whitelist.none() );

                return value;
        }
}
```

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

Risk of using projects 3

- Java based
- Version 8.0, long lasting
- Includes lessons and hacks

- Collection of DevOps-driven applications, specifically designed to showcase security catastrophes
- Micro services and containerization

- JavaScript based
- Intentionally insecure web app
- Encompasses the entire OWASP Top Ten and other severe security flaws

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

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

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
- Secure coding examples in multiple languages

Hands-on training

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

+1

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 the different training apps
- Schedule a hack-a-thon where teams gather together and work on the vulnerable apps in teams and learn from each other

Process and Measurement

Application Security Verification Standard

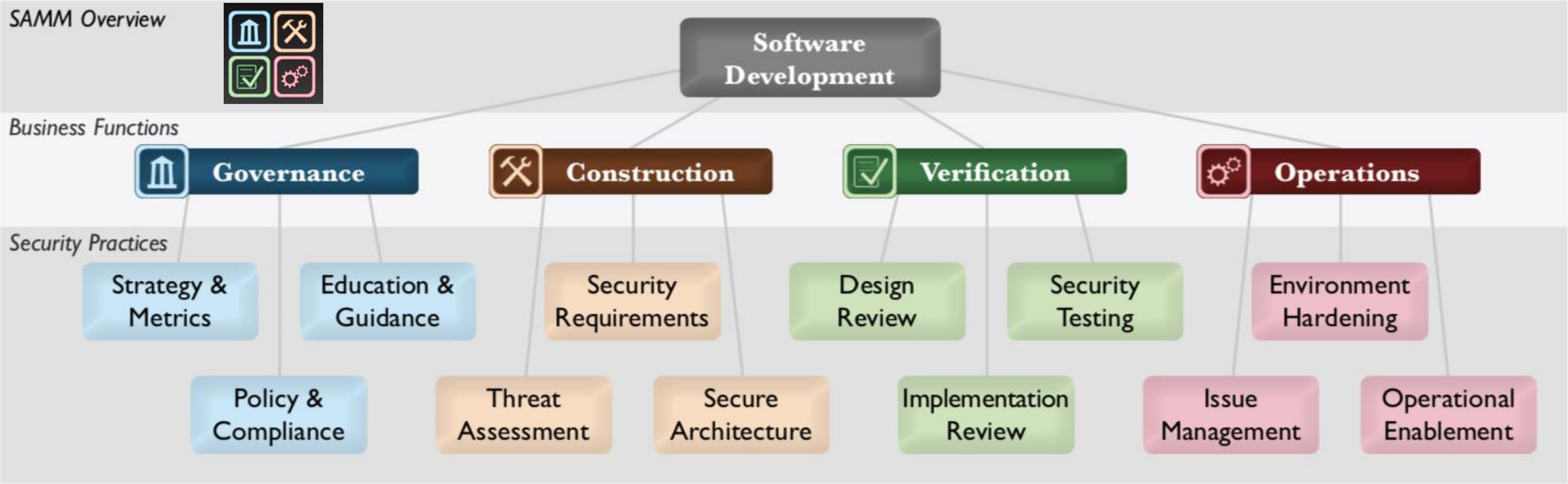
OWASP | Testing Guide
Open Web Application Security Project

CODE
REVIEW
GUIDE

Application Threat Modeling

Measurement

Risk of using project 1



SAMM Overview

Software Development

Business Functions

Governance

Construction

Verification

Operations

Security Practices

Strategy & Metrics

Policy & Compliance

Education & Guidance

Threat Assessment

Security Requirements

Secure Architecture

Design Review

Implementation Review

Security Testing

Issue Management

Environment Hardening

Operational Enablement

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

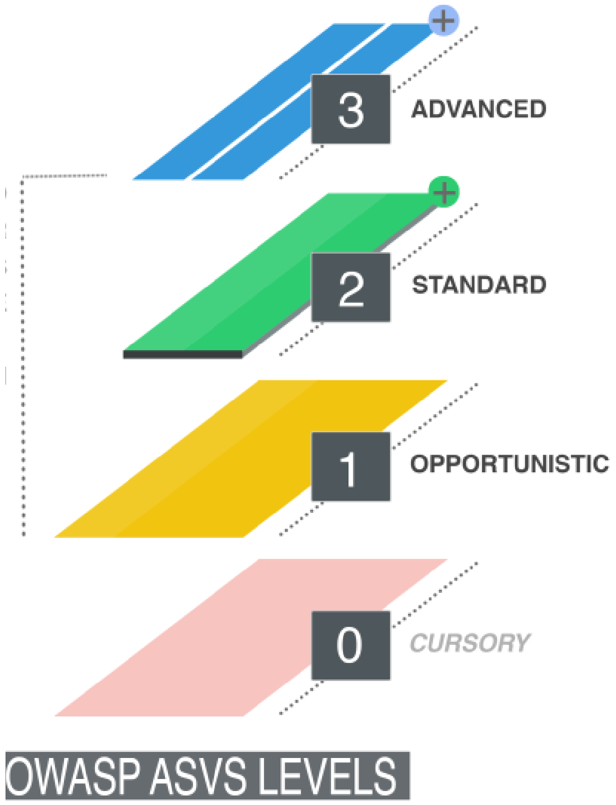
Process

Knowledge

Risk of using project 0

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	

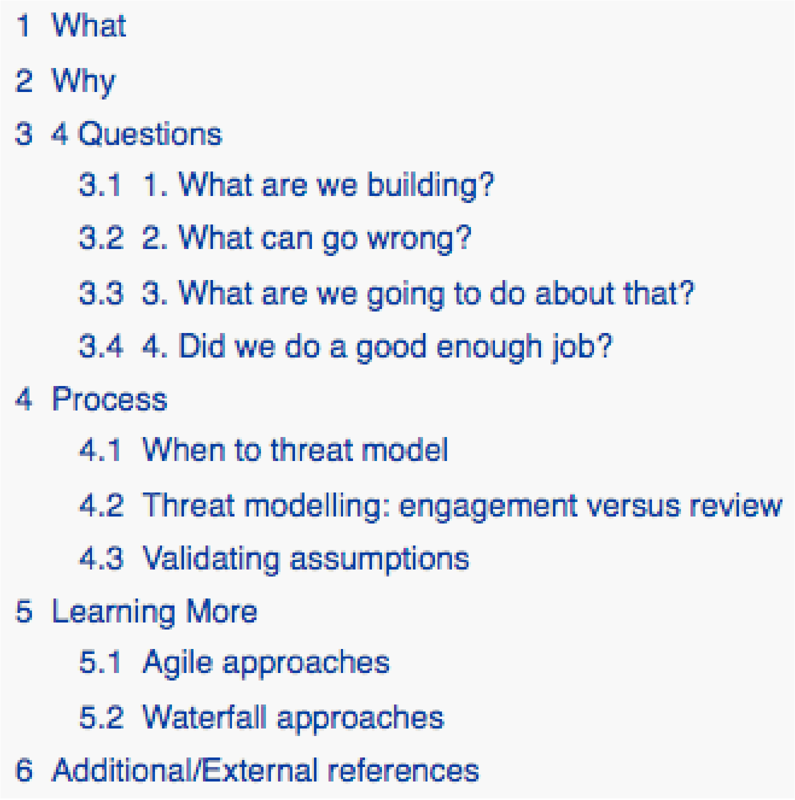
Application Security Verification Standard



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

Process

Risk of using project 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

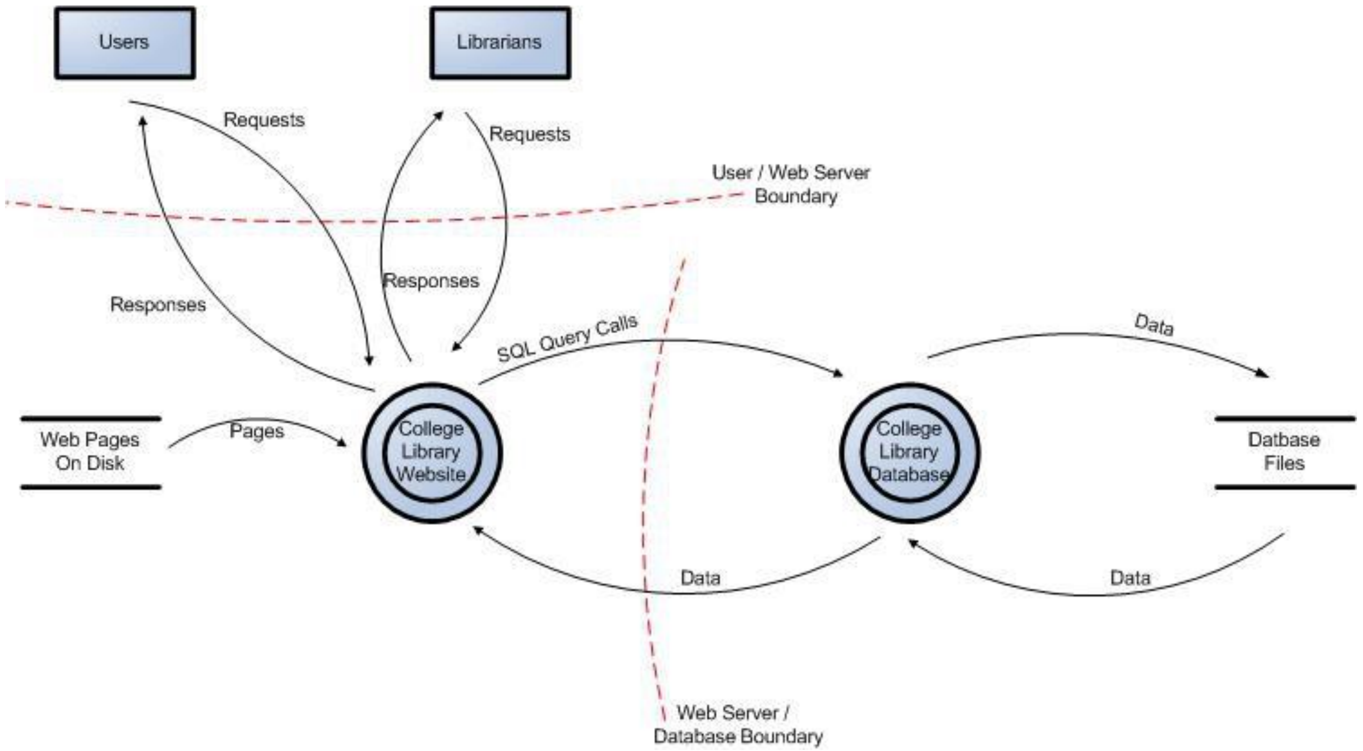
- 4.1 When to threat model
- 4.2 Threat modelling: engagement versus review
- 4.3 Validating assumptions

5 Learning More

- 5.1 Agile approaches
- 5.2 Waterfall approaches

6 Additional/External references

Application Threat Modeling

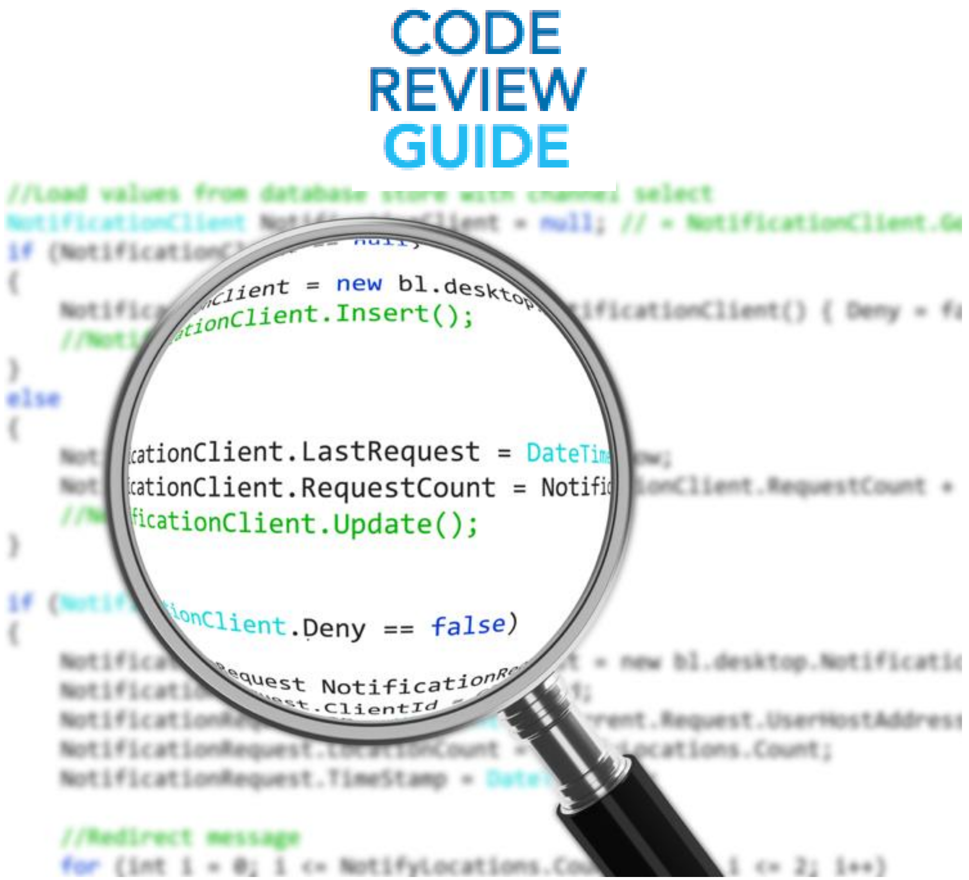


https://www.owasp.org/index.php/Application_Threat_Modeling

Process

Risk of using project 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

Process

Knowledge Base

Risk of using project 1

Principles and techniques of testing

Phases of a test

Configuration and deployment management testing

Identity management testing

Authentication testing

Authorization testing

Session management testing

Input validation testing

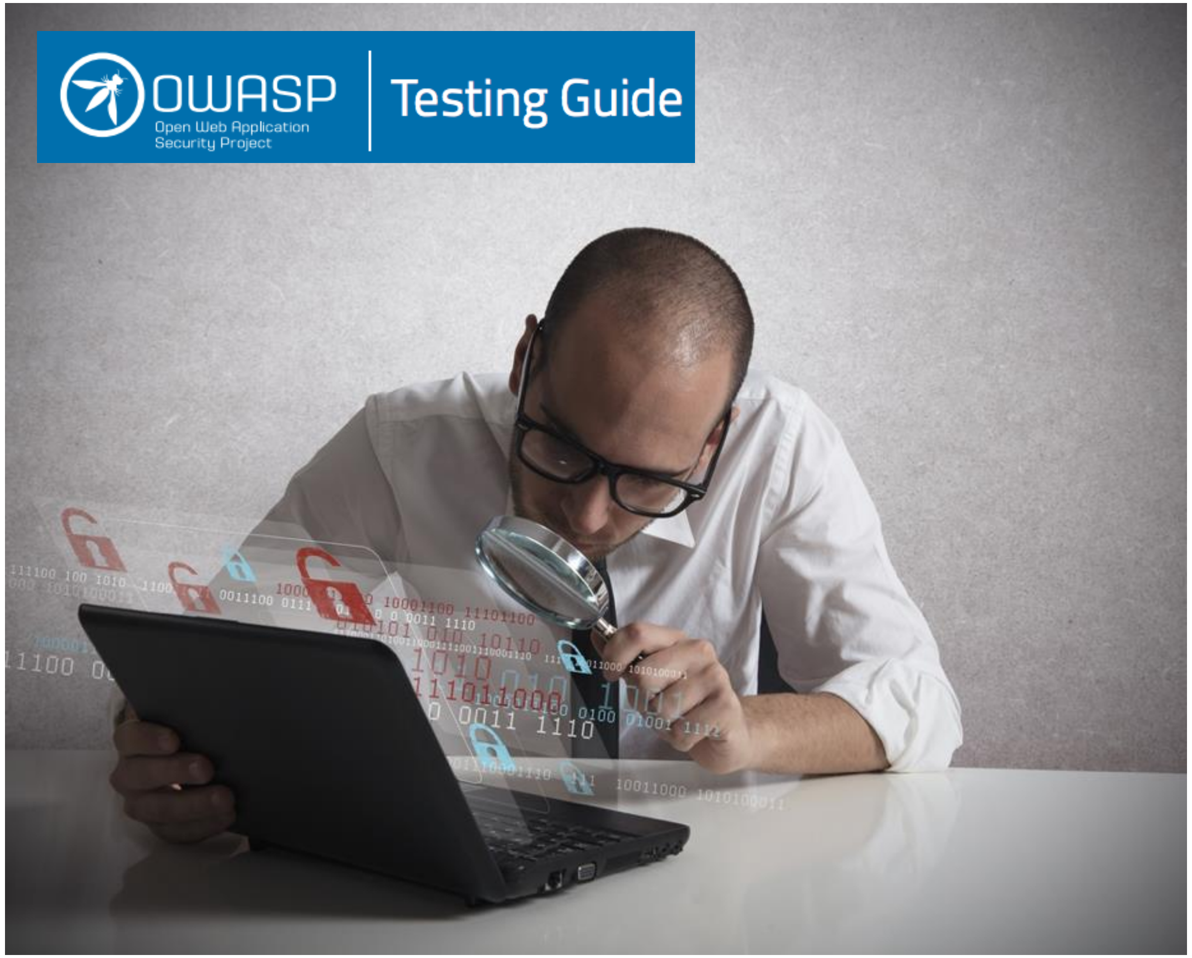
Testing for error handling

Testing for weak crypto

Business logic testing

Client side testing

Reporting



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

+1

Measurement

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

+.5

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

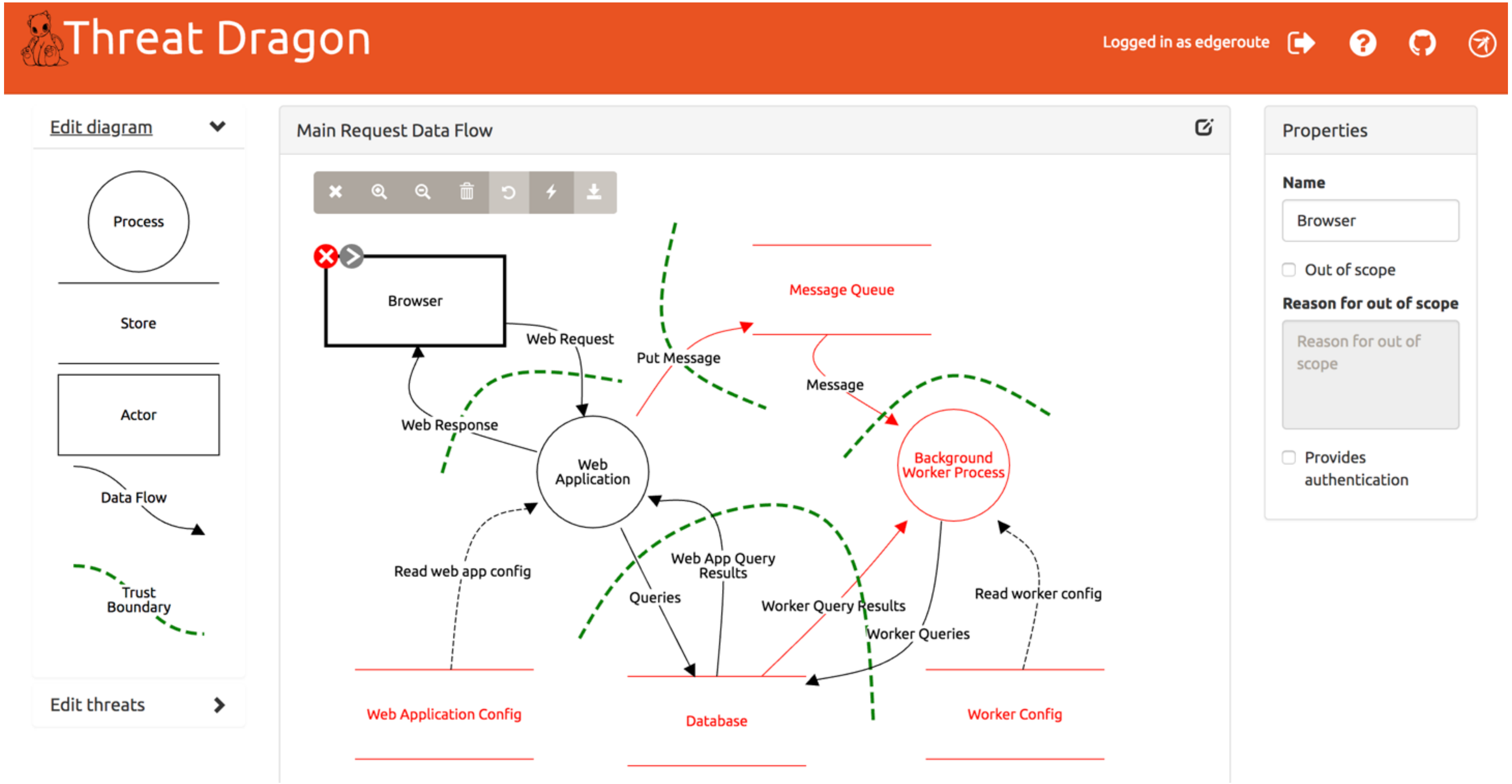
OWASP
ModSecurity
Core Rule Set
THE 1ST LINE OF DEFENSE

DEPENDENCY-CHECK

OWASP AppSec Europe
London 2nd-6th July 2018

Design

Risk of using project 5



Properties

Name

Browser

□ Out of scope

Reason for out of scope

Reason for out of scope

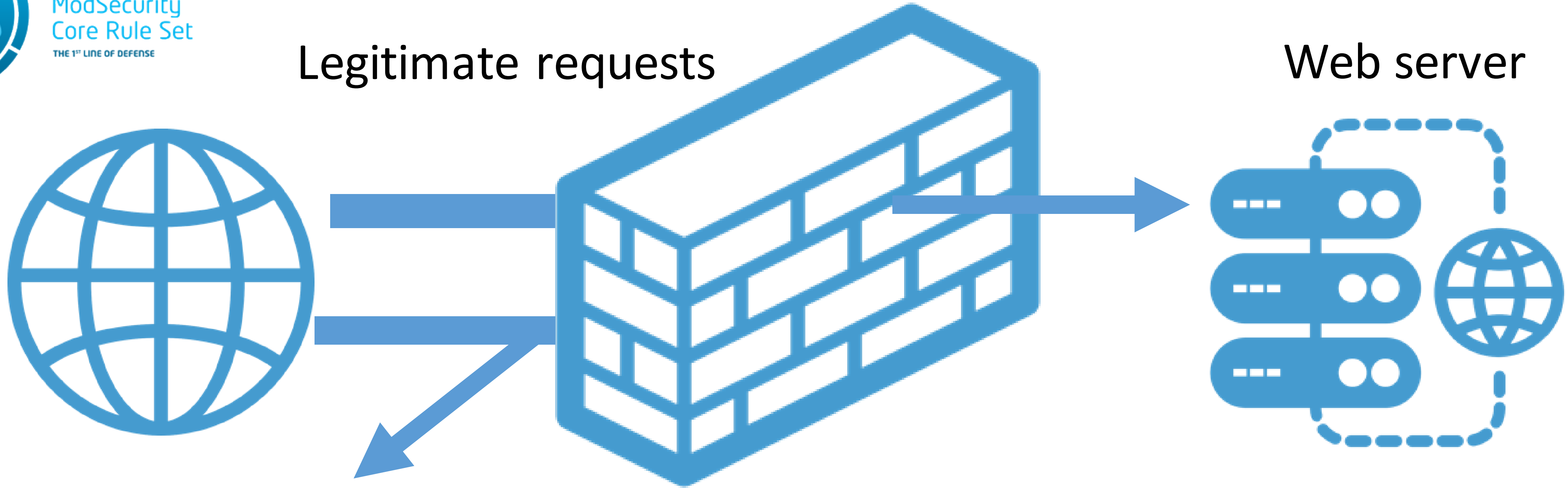
□ Provides authentication

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

Infrastructure

Risk of using project 1

OWASP ModSecurity Core Rule Set THE 1ST LINE OF DEFENSE

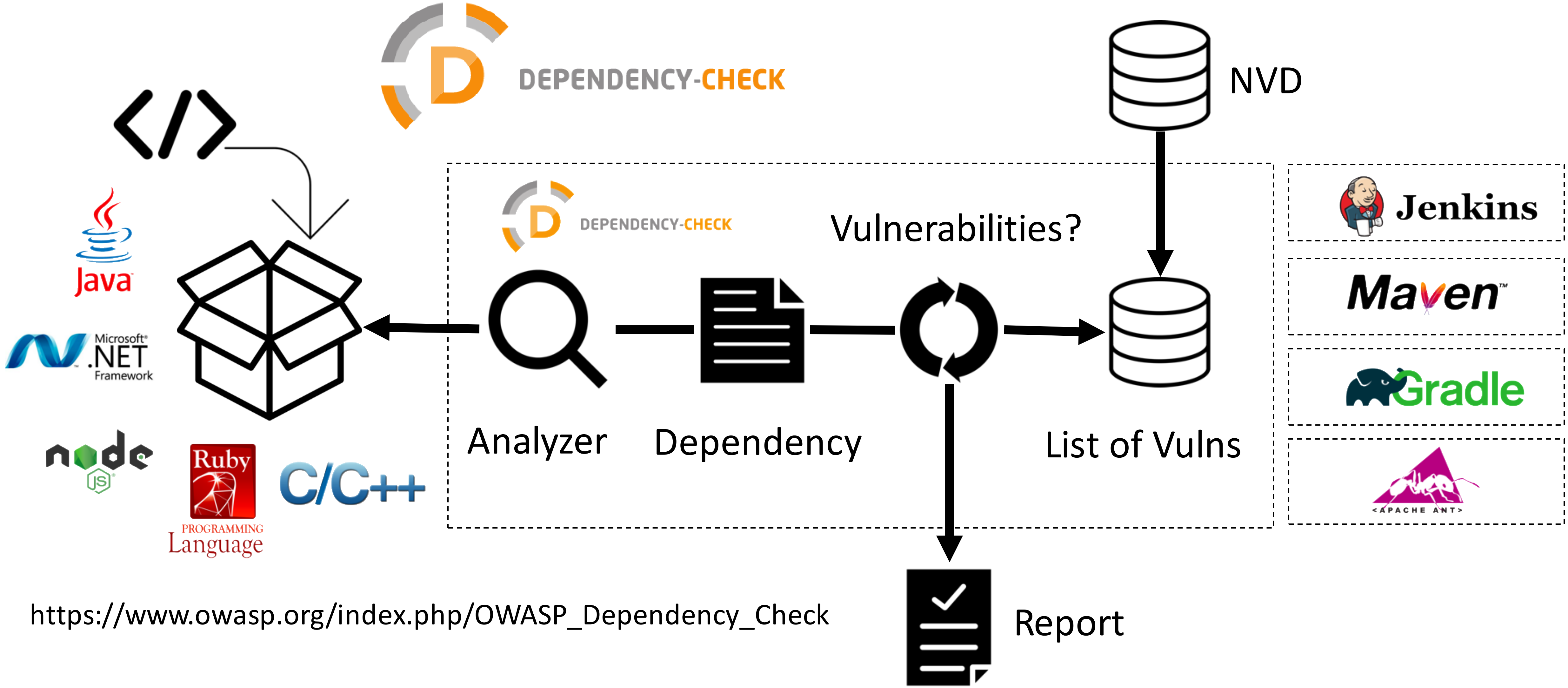


ModSecurity w/
Core Rule Set

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

Infrastructure

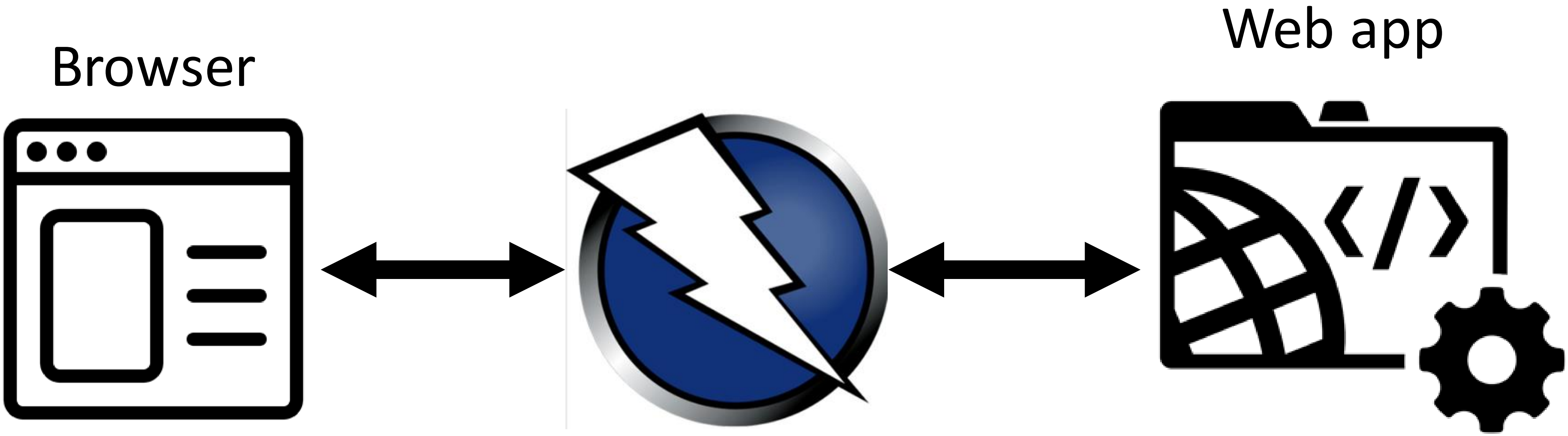
Risk of using project 3



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

Infrastructure

Risk of using project 2



Browser

Web app

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

Design

- Threat dragon provides a new, web based approach to capturing threats that will reach Enterprise status if it delivers on the roadmap

+1

Infrastructure

- CRS provides a true WAF solution
- Dependency check identifies vulnerable 3rd party software
- ZAP provides DAST, and plugs in to any dev methodology

+1

Tools: getting started

Design

- Use threat dragon as the tool to teach threat modeling and scale it across your development teams
  - Partner with application threat modeling knowledge

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

Awareness and education

Process and measurement

Tools

Awareness

Knowledge

Hands-on training

+1

Knowledge

Hands-on training

+1.5

Design

Infrastructure

+2

The OWASP stack as an AppSec program

Tools

- Design
- Infrastructure

DEPENDENCY-CHECK

OWASP ModSecurity Core Rule Set
THE 1st LINE OF DEFENSE

Process and measurement

- Process
- Measurement

Application Security Verification Standard

Application Threat Modeling

OWASP Open Web Application Security Project | Testing Guide

CODE REVIEW GUIDE

Awareness and education

- Awareness
- Knowledge
- Hands-on training

OWASP Top 10 - 2017
The Ten Most Critical Web Application Security Risks

Security Knowledge Framework

OWASP Cheat Sheets

OWASP Automated Threat Handbook Web Applications

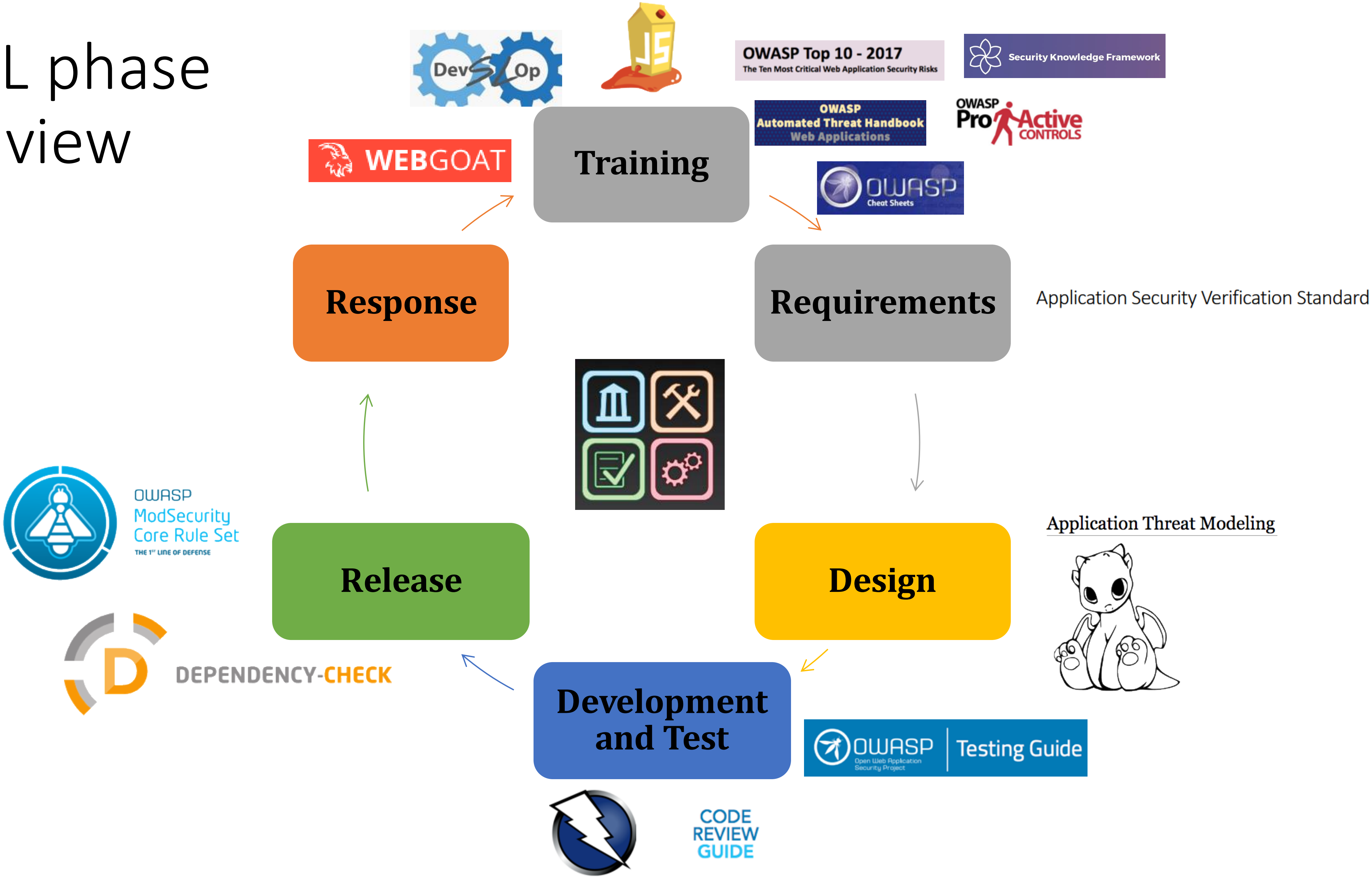
OWASP Pro Active CONTROLS

DevSlop

WEBGOAT

Security Community

SDL phase view



Training

Requirements

Design

Development and Test

Release

Response

DevSlop
OWASP Top 10 - 2017
The Ten Most Critical Web Application Security Risks
Security Knowledge Framework
OWASP Automated Threat Handbook Web Applications
OWASP ProActive CONTROLS
OWASP Cheat Sheets
WEBGOAT
Application Security Verification Standard
Application Threat Modeling
OWASP Open Web Application Security Project | Testing Guide
CODE REVIEW GUIDE
DEPENDENCY-CHECK
OWASP ModSecurity Core Rule Set
THE 1st LINE OF DEFENSE

Final thoughts for an AppSec program on the cheap

1. Use OpenSAMM to assess current program state 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 the projects available 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 accordingly 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