



**OWASP**

# OWASP

- **Open Web Application Security Project**
  - Free and open software security community
  - 501(c)(3) Nonprofit organization
- Core purpose
  - Be the thriving global community that drives visibility and evolution in the safety and security of the world's software

# Core Values

- **OPEN** Everything at OWASP is radically transparent from our finances to our code.
- **INNOVATION** OWASP encourages and supports innovation and experiments for solutions to software security challenges.
- **GLOBAL** Anyone around the world is encouraged to participate in the OWASP community.
- **INTEGRITY** OWASP is an honest and truthful, vendor neutral, global community.

# Principles

- Free & Open
- Governed by rough consensus & running code
- Abide by a [code of ethics](#)
- Not-for-profit
- Not driven by commercial interests
- Risk based approach

# OWASP Projects

# OWASP Projects




An OWASP project is a collection of related tasks that have a defined roadmap and team members.

Project Type	Examples
Tool	<a href="#">Zed Attack Proxy</a> , <a href="#">Dependency Check</a> , <a href="#">DefectDojo</a> , <a href="#">Juice Shop</a>
Code	<a href="#">ModSecurity Core Rule Set</a> , <a href="#">Java HTML Sanitizer</a> , <a href="#">Security Logging Project</a> , <a href="#">AppSensor</a>
Documentation	<a href="#">OWASP Top 10</a> , <a href="#">Application Security Verification Standard (ASVS)</a> , <a href="#">OWASP 24/7 Podcast</a> , <a href="#">Cornucopia</a>

# OWASP Communities

Community	Advancing the state of security in the area of...
Builders	...application development
Breakers	...security testing
Defenders	...application defense, including the tools and techniques that enable the detection and response to application layer attacks

# Project Lifecycle

Level	Icon	Description
Incubator		OWASP Incubator projects represent the experimental playground where projects are still being fleshed out, ideas are still being proven, and development is still underway.
Labs		OWASP Labs projects represent projects that have produced an OWASP reviewed deliverable of value.
Flagship		The OWASP Flagship designation is given to projects that have demonstrated strategic value to OWASP and application security as a whole.





## Determining Lab Status

It is essential for an OWASP Labs project to have:

- A version number with a clear release schedule
- GitHub source control and a public issue tracking system
- Stable build and release
- Instructions on how to use and build the project properly



# Determining Flagship Status

It is essential for an OWASP Flagship project to have:

- Considerable number of users and contributors
- Considerable number of commits and improvements in a time span of at least two years
- A unique approach or proposition in application security
- Exposure through security conferences
- Use and acceptance by the community
- Being used as reference in books and other resources

# OWASP Chapters

# OWASP Chapters

OWASP Chapters exist to raise awareness of the OWASP mission, making application security visible, at the local level.

## OWASP German Chapter

There is one Chapter for Germany in total which is complemented by a so-called [OWASP Stammtisch](#) each in several metropolitan areas such as [Hamburg](#), Munich, Frankfurt, Stuttgart or Karlsruhe.



# Mandatory Chapter Rules

- Organize free and open meetings
- Hold a minimum of 4 chapter meetings or events each year
- Give official meeting notice through the wiki, chapter mailing list, and OWASP Calendar
- Abide by OWASP principles and the code of ethics
- Protect the privacy of the chapter's local contacts
- Maintain vendor neutrality (act independently)
- Spend any chapter funds in accordance with the OWASP goals, code of ethics, and principles

# OWASP Top 10



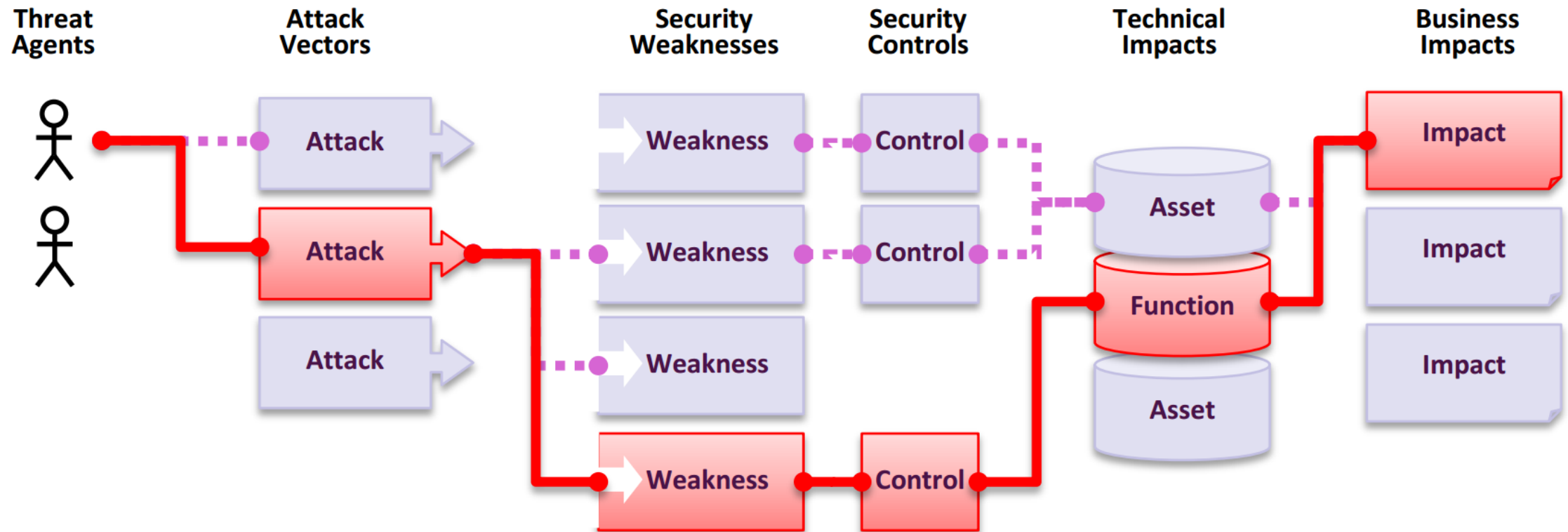
# OWASP Top 10

Builders

Defenders

1	Injection	6	Security Misconfiguration
2	Broken Authentication	7	Cross-Site-Scripting (XSS)
3	Sensitive Data Exposure	8	Insecure Deserialization
4	XML External Entities	9	Using Components with Known Vulnerabilities
5	Broken Access Control	10	Insufficient Logging & Monitoring

# Application Security Risks





# Risk Rating Table

Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	SEVERE: 3	App / Business Specific
	AVERAGE: 2	COMMON: 2	AVERAGE: 2	MODERATE: 2	
	DIFFICULT: 1	UNCOMMON: 1	DIFFICULT: 1	MINOR: 1	

**i** Based on the *OWASP Risk Rating Methodology*

# Risk Calculation Example

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App Specific	Exploitability: 3	Prevalence: 3	Detectability: 3	Technical: 2	Business ?
	3	3	3		
	Likelihood Rating: 3.0  (Average of Exploitability, Prevalence and Detectability)			* 2	
	Risk Ranking: 6.0  (Likelihood * Impact)				

# Top 10 Risk Factor Summary

Risk	Threat Agents	Attack Vectors (Exploitability)	Security Weakness (Prevalence)	Security Weakness (Detectability)	Impacts (Technical)	Impacts (Business)	Score
<a href="#">A1:2017-Injection</a>	App Specific	EASY: 3	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	8.0
<a href="#">A2:2017-Broken Authentication</a>	App Specific	EASY: 3	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	7.0
<a href="#">A3:2017-Sensitive Data Exposure</a>	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	SEVERE: 3	App Specific	7.0
<a href="#">A4:2017-XML External Entities (XXE)</a>	App Specific	AVERAGE: 2	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	7.0
<a href="#">A5:2017-Broken Access Control</a>	App Specific	AVERAGE: 2	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	6.0
<a href="#">A6:2017-Security Misconfiguration</a>	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
<a href="#">A7:2017-Cross-Site Scripting (XSS)</a>	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
<a href="#">A8:2017-Insecure Deserialization</a>	App Specific	DIFFICULT: 1	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	5.0
<a href="#">A9:2017-Vulnerable Components</a>	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	MODERATE: 2	App Specific	4.7
<a href="#">A10:2017-Insufficient Logging&amp;Monitoring</a>	App Specific	AVERAGE: 2	WIDESPREAD: 3	DIFFICULT: 1	MODERATE: 2	App Specific	4.0

# Some(!) Additional Risks to Consider

Cross-Site Request Forgery (CSRF)	Unvalidated Forward and Redirects
Uncontrolled Resource Consumption ('Resource Exhaustion', 'AppDoS')	Improper Control of Interaction Frequency (Anti-Automation)
Unrestricted Upload of File with Dangerous Type	Inclusion of Functionality from Untrusted Control Sphere (3rd Party Content)
User Interface (UI) Misrepresentation of Critical Information (Clickjacking etc.)	Server-Side Request Forgery (SSRF)

# Other Resources on AppSec

- **SANS** Software Security Community
  - [CWE/SANS TOP 25 Most Dangerous Software Errors](#)
  - [Securing Web Application Technologies \[SWAT\] Checklist](#)
- **CWE** Common Weakness Enumeration
  - Community-developed list of common software security weaknesses



# OWASP Juice Shop



# OWASP Juice Shop

Builders

Breakers

Defenders

OWASP Juice Shop is probably the most modern and sophisticated insecure web application! It can be used in security trainings, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire OWASP Top Ten along with many other security flaws found in real-world applications!



# Main Selling Points


- **Easy-to-install:** Choose between node.js, Docker and Vagrant to run on Windows/Mac/Linux
- **Self-contained:** Additional dependencies are pre-packaged or will be resolved and downloaded automatically
- **Self-healing:** The simple SQLite database is wiped and regenerated from scratch on every server startup
- **Gamification:** The application notifies you on solved challenges and keeps track of successfully exploited vulnerabilities on a Score Board



- **CTF-support:** Challenge notifications contain a customizable flag code for your own Capture-The-Flag events
- **Re-branding:** Fully customizable business context and look & feel
- **Free and Open source:** Licensed under the MIT license with no hidden costs or caveats



# Installation

- Individual local instance per student
- Runs on node.js, Docker, Vagrant and in the 

## Hacking Rules

- Do **not** look at the source code on GitHub
- Do **not** look at GitHub issues, PRs etc.
- Do **not** cheat (with online tutorials or walkthroughs) before trying
- Report problems during exercises immediately

# Official Companion Guide

[Pwning OWASP Juice Shop](#) [...] will give you a complete overview of the vulnerabilities found in the application including hints how to spot and exploit them. In the appendix you will even find complete step-by-step solutions to every challenge. The ebook is published under [CC BY-NC-ND 4.0](#) and is available **for free online-readable**. The latest officially released edition is **available for free on LeanPub** in PDF, Kindle and ePub format.



THIS IS THE OFFICIAL COMPANION GUIDE TO THE OWASP JUICE SHOP APPLICATION. BEING A WEB APPLICATION WITH A VAST NUMBER OF INTENDED SECURITY VULNERABILITIES, THE OWASP JUICE SHOP IS SUPPOSED TO BE THE OPPOSITE OF A BEST PRACTICE OR TEMPLATE APPLICATION FOR WEB DEVELOPERS: IT IS AN AWARENESS, TRAINING, DEMONSTRATION AND EXERCISE TOOL FOR SECURITY RISKS IN MODERN WEB APPLICATIONS. THE OWASP JUICE SHOP IS AN OPEN-SOURCE PROJECT HOSTED BY THE NON-PROFIT OPEN WEB APPLICATION SECURITY PROJECT (OWASP) AND IS DEVELOPED AND MAINTAINED BY VOLUNTEERS.

BJÖRN KIMMINICH HAS OVER TWO DECADES OF PROGRAMMING EXPERIENCE WITH EXPERTISE ON SOFTWARE SUSTAINABILITY, CLEAN CODE AND TEST AUTOMATION AS WELL AS APPLICATION SECURITY. HE IS THE PROJECT LEADER OF THE OWASP JUICE SHOP AND MEMBER OF THE GERMAN OWASP CHAPTER BOARD.



# Exercise 1.1

## Install the OWASP Juice Shop **v9.x**

1. Open <https://github.com/bkimminich/juice-shop#setup>
2. Follow the instructions for one method out of
  - [From Sources](#)
  - [Packaged Distributions](#) (❤️ on university PCs)
  - [Docker Container](#)
  - [Vagrant](#)

# Exercise 1.2

## Happy path shopping tour

1. Register a user account at your local Juice Shop
2. Browse the inventory and purchase some products
3. Try out all other functionality you find in the application

# Exercise 1.3

## Score Board

1. Find the hidden Score Board in the Juice Shop (★)

## Exercise 1.4 (*optional*)

### Transfer your hacking progress

1. Open your browser's developer tools ( `F12` in Chrome/Firefox)
2. Find the cookie `continueCode` and copy its value to your other computer
3. Install OWASP Juice Shop on your other computer and launch it
4. `F12` into the developer tools and create the cookie `continueCode` with the value from your first computer
5. Restart the Juice Shop server