Build secure software with OWASP tools and guides
Martin Knobloch
+10 years developer experience
+15 years information security experience

Dutch OWASP Chapter Leader since 2007
OWASP Foundation Board of Directors since 2017

Global AppSec Strategist @ Micro Focus / Fortify

Contact:

martin.knobloch@owasp.org

https://twitter.com/knoblochmartin
https://www.linkedin.com/in/martin-knobloch
https://xing.to/knoblochmartin
Who is the OWASP® Foundation?

The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open source software projects, hundreds of local chapters worldwide, tens of thousands of members, and training and professional and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

- Tools and Resources
- Community and Networking
- Education & Training

For nearly two decades, corporations, foundations, developers, and volunteers have supported the OWASP Foundation and its work. Donate, Join, or become a Corporate Member today.

Project Spotlight: Mobile Security Testing Guide

The OWASP® Mobile Security Testing Guide (MSTG) is a comprehensive manual for mobile app security model and test guide, security requirements for mobile apps, while the MSTG serves as a baseline for mobile security testing and as a template for automating security tests during or after development. It includes the MSTG, the Mobile Security Testing Playground as a collection of iOS and Android mobile apps that are intentionally built insecure. These apps are available to download, reverse-engineer, and understand the vulnerabilities.

Featured Chapter: Bay Area

Headed up by some of the best tech companies in the world, the Bay Area chapter is one of the Foundation’s largest and most active. This month they are hosting a Hacker Day and interactive workshops in San Francisco at Oracle Engineering and in South Bay at eBay. Usually, the agenda includes interactive, fun, and informative talks, lots of interesting people to meet, and great food. The Bay Area chapter also participates in planning NapaCon California.
• to make application security "visible," so that people and organizations can make informed decisions about application security risks
<table>
<thead>
<tr>
<th>OWASP Top 10 – 2013 (Previous)</th>
<th>OWASP Top 10 – 2017 (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 – Injection</td>
<td>A1 – Injection</td>
</tr>
<tr>
<td>A2 – Broken Authentication and Session Management</td>
<td>A2 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A3 – Cross-Site Scripting (XSS)</td>
<td>A3 – Cross-Site Scripting (XSS)</td>
</tr>
<tr>
<td>A5 – Security Misconfiguration</td>
<td>A5 – Security Misconfiguration</td>
</tr>
<tr>
<td>A6 – Sensitive Data Exposure</td>
<td>A6 – Sensitive Data Exposure</td>
</tr>
<tr>
<td>A7 – Missing Function Level Access Control - Merged with A4</td>
<td>A7 – Insufficient Attack Protection (NEW)</td>
</tr>
<tr>
<td>A8 – Cross-Site Request Forgery (CSRF)</td>
<td>A8 – Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A9 – Using Components with Known Vulnerabilities</td>
<td>A9 – Using Components with Known Vulnerabilities</td>
</tr>
<tr>
<td>A10 – Unvalidated Redirects and Forwards - Dropped</td>
<td>A10 – Underprotected APIs (NEW)</td>
</tr>
</tbody>
</table>

https://owasp.org/www-project-top-ten/
Agile – CI/CD - DevOps

Agile

- focuses on processes
- highlighting change
- while accelerating delivery

CI/CD

- focuses on software-defined life cycles
- highlighting tools
- that emphasize automation

DevOps

- focuses on culture
- highlighting roles
- that emphasize responsiveness

With thanks to John Steven
OWASP
The Open Web Application Security Project

Faster towards production
..but
https://owasp.org/www-project-samm/
C1: Define Security Requirements
C2: Leverage Security Frameworks and Libraries
C3: Secure Database Access
C4: Encode and Escape Data
C5: Validate All Input
C6: Implement Digital Identity
C7: Enforce Access Controls
C8: Protect Data Everywhere
C9: Implement Security Logging and Monitoring
C10: Handle All Errors and Exceptions

https://owasp.org/www-project-proactive-controls/
The OWASP Cheat Sheet Series was created to provide a concise collection of high value information on specific application security topics. These cheat sheets were created by various application security professionals who have expertise in specific topics.

https://cheatsheetseries.owasp.org/
https://owasp.org/www-project-cheat-sheets/
V1: Architecture, Design and Threat Modeling Requirements
V2: Authentication Verification Requirements
V3: Session Management Verification Requirements
V4: Access Control Verification Requirements
V5: Validation, Sanitization and Encoding Verification Requirements
V6: Stored Cryptography Verification Requirements
V7: Error Handling and Logging Verification Requirements
V8: Data Protection Verification Requirements
V9: Communication Verification Requirements
V10: Malicious Code Verification Requirements
V11: Business Logic Verification Requirements
V12: File and Resources Verification Requirements
V13: API and Web Service Verification Requirements
V14: Configuration Verification Requirements

https://owasp.org/www-project-application-security-verification-standard/
### Requirements

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.4</td>
<td>Verify that access to sensitive records is protected, such that only authorized objects or data is accessible to each user (for example, protect against users tampering with a parameter to see or alter another user’s account).</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.5</td>
<td>Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.8</td>
<td>Verify that access controls fail securely.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.9</td>
<td>Verify that the same access control rules implied by the presentation layer are enforced on the server side.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.10</td>
<td>Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.11</td>
<td>Verify that there is a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>1.0</td>
</tr>
<tr>
<td>4.12</td>
<td>Verify that all access control decisions can be logged and all failed decisions are logged.</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>2.0</td>
</tr>
</tbody>
</table>

[https://owasp.org/www-project-application-security-verification-standard/](https://owasp.org/www-project-application-security-verification-standard/)
https://owasp.org/www-project-cornucopia/
Threat Dragon

https://owasp.org/www-project-threat-dragon/
Detect possible threats in your application
In pre-development detect possible threats based on the processing functions on your application.

Run OWASP ASVS Checklists
Harden your application functions in post-development by running OWASP ASVS checklists, complete with feedback and solutions.

Learn about threats and vulnerabilities in the SKF knowledge base
An extensive library of common hacks and exploits, learn the hacker mindset and keep your project secure.

Learn to code secure from best practice code examples
An extensive library of code examples for a wide range of functions, beautifully commented.

https://www.securityknowledgeframework.org/
https://owasp.org/www-project-security-knowledge-framework/
Information Gathering
Configuration and Deploy Management Testing
Identity Management Testing
Authentication Testing
Authorization Testing
Session Management Testing
Input Validation Testing
Testing for Error Handling
Testing for Weak Cryptography
Business Logic Testing
Client-Side Testing
API Testing

https://owasp.org/www-project-web-security-testing-guide/
OWASP
The Open Web Application Security Project

https://owasp.org/www-project-zap/
https://www.zaproxy.org/
OWASP
The Open Web Application Security Project

https://owasp.org/www-project-zap/
https://www.zaproxy.org/
Stages of an AppSec Pipeline

The first stage of an AppSec Pipeline which handles inbound requests of the AppSec program. These can be new apps, existing apps that have never been assessed, apps which have been assessed before or retesting of previous security findings. These tools aim to tame the inflow of work into the AppSec Pipeline.

The second stage of an AppSec Pipeline which prioritizes inbound requests and assesses their testing needs based on the risk level. The more risky the app, the more activities are assigned. These tools aim to provide automation and orchestration to reduce the startup time of the testing stage.

The third stage of an AppSec Pipeline which runs one or more tests in parallel to assess the security posture of an application. Ideally, these tests or at least their setup should be automated. Priority should be given to tools that can be run programmatically and produce results with few false positives.

The forth and final stage of an AppSec Pipeline which collects and normalizes the data created during testing. Any duplicate findings should be removed so that the same issue found by multiple tools is only reported once. Here we link to issue tracking systems, produce reports, and otherwise provide data for stakeholders.

https://owasp.org/www-project-appsec-pipeline/
https://www.appsecpipeline.org/
Rugged Devops - AppSec Pipeline Template

https://owasp.org/www-project-appsec-pipeline/
https://www.appsecpipeline.org/
https://owasp.org/www-project-defectdojo/
https://www.defectdojo.org/
https://owasp.org/www-project-webgoat/
https://owasp.org/www-project-juce-shop/
https://owasp.org/www-project-integration-standards/
https://www.youtube.com/playlist?list=PLUKo5k_oSrfOTI27gUmk2o-NBKvkTGw0T