

IMPROVING THE SECURITY OF SOFTWARE

OWASP Foundation

The OWASP Foundation

- We are a **Global not-for-profit charitable** organisation
- Vendor-Neutral Community
- **Collective Wisdom of the Best Minds in Application Security Worldwide**
- Provide **free** tools, guidance, documentation

We are all VOLUNTEERS!



owasp.org

45,000+ OWASP volunteers worldwide

World Wide

- 207 local Chapters in 56 countries... and counting!



OWASP.ORG

**Annually, about
seven million
unique visitors
use owasp.org**



The screenshot shows a web browser displaying the OWASP.org homepage. The address bar shows "owasp.org". The header features the OWASP logo and navigation links for PROJECTS, CHAPTERS, EVENTS, and ABOUT. A search bar and donation buttons are also present. A banner at the top encourages registration for "Global AppSec 2020". Below the banner, there is a large image of a man speaking to an audience in a conference setting. To the right of the image, a section titled "Who is the OWASP® Foundation?" provides information about the organization's mission and work. A bulleted list below this section highlights "Tools and Resources", "Community and Networking", and "Education & Training". At the bottom of the page, a message thanks supporters and provides links for "Donate", "Join", and "Corporate Member".

Register now for Global AppSec 2020. Great keynotes, training, over 60 education sessions, and more.

OWASP PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org

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 leading educational 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.

IMPROVING SOFTWARE SECURITY, WORLDWIDE

OWASP Projects

- 189 Projects including 20 Flagship Projects

Flagship Projects



- OWASP Amass
- OWASP Application Security Verification Standard
- OWASP Cheat Sheet Series
- OWASP CSRFGuard
- OWASP Defectdojo
- OWASP Dependency-Check
- OWASP Dependency-Track
- OWASP Juice Shop
- OWASP Maryam
- OWASP Mobile Security Testing Guide
- OWASP Mobile Top 10
- OWASP ModSecurity Core Rule Set
- OWASP OWTF
- OWASP SAMM
- OWASP security Knowledge Framework
- OWASP Security Qualitative Metrics
- OWASP Security Shepherd
- OWASP Top Ten
- OWASP Web Security Testing Guide
- OWASP ZAP

Lab Projects



- OWASP AntiSamy
- OWASP API Security Project
- OWASP Attack Surface Detector
- OWASP Automated Threats to Web Applications
- OWASP Benchmark
- OWASP Code Pulse
- OWASP Cornucopia
- OWASP Enterprise Security API (ESAPI)
- OWASP Internet of Things
- OWASP Java HTML Sanitizer
- OWASP mobile security
- OWASP Proactive Controls
- OWASP Secure Coding Dojo
- OWASP Snakes And Ladders
- OWASP Top 10 Privacy Risks
- OWASP TorBot
- OWASP WebGoat

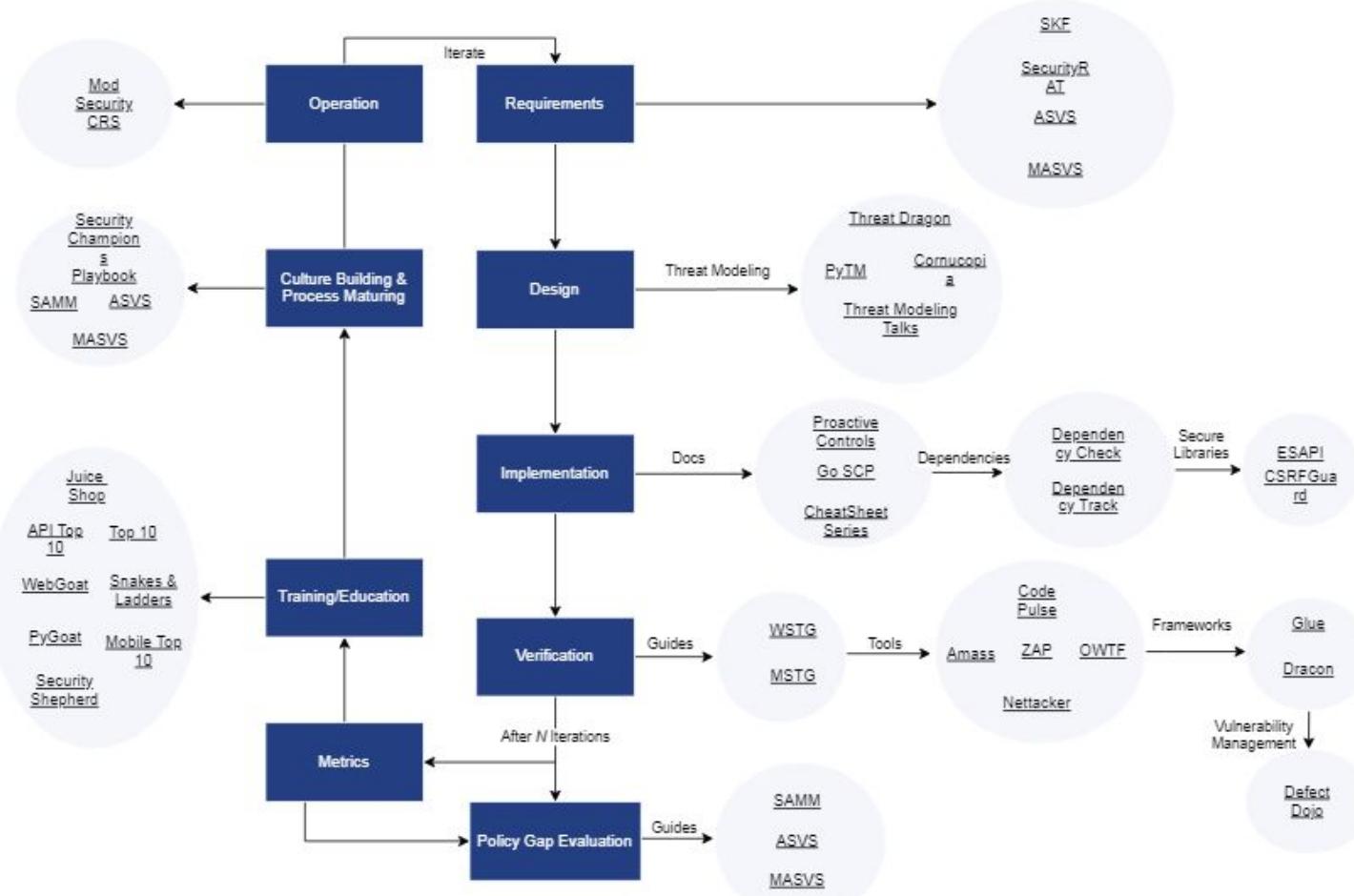
Incubator Projects



List: <https://owasp.org/projects/>

Application Security Wayfinder

Brought to you by the Integration standards project
Linking requirements and guidance across standards through the Common Requirement Enumeration.



<https://owasp.org/www-project-integration-standards/>

OWASP Community Events

- Our mission is to make application security **visible** so that people and organisations can make **informed decisions** about application security risks.
- Meetings are free to attend (free drinks & food included)
- Meetings are usually 1-2 hour seminars or workshops



Session at Global AppSec Amsterdam

Canadian Chapters

- [OWASP Calgary](#)
- [OWASP Montreal](#)
- [OWASP Moncton](#)
- [OWASP Ottawa](#)
- [OWASP Quebec City](#)
- [OWASP Toronto](#)
- [OWASP Vancouver](#)

<https://owasp.org/chapters/>



It's all for free

- Everyone is **free** to participate in OWASP and **all** of our materials are available under a **free** and **open** software license.
- All OWASP events (*except conferences*) are free to attend by both members and non-members of OWASP - and can be attended by anyone who is interested in Application Security and Cyber Security in general.



Member Lounge at OWASP Conference

Premier members (donate \$20,000/year):



Signal Sciences



New Corporate Members



hackerone

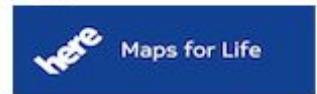
facebook



kenna

Contributing Members

These corporate members support OWASP at the \$5,000 USD level annually.



Keep In Touch

Join an OWASP Mailing List:

<https://groups.google.com/a/owasp.com>

Follow us on Twitter



“Like” us on Facebook

www.facebook.com/groups/owaspfoundation



Slack: owasp.slack.com #owasp-community



Watch us on **YouTube**:

YouTube.com/OWASPGLOBAL



www.meetup.com/pro/OWASP

Visit the **OWASP** website

<https://owasp.org>

Introduction to OWASP Projects

(a small sample)

OWASP Zed Attack Proxy

More than just a proxy

Intro to OWASP Zed Attack Proxy (ZAP)

OWASP Flagship Project!

Free and Open source!
Apache 2 License

An HTTP proxy and DAST Tool for
Breakers and Builders Alike!



ZAP for Testers

Untitled Session - OWASP ZAP 2.9.0

File Edit View Analyse Report Tools Import Online Help

Standard Mode

Sites Scripts

Quick Start Request Break Script Console Response

Header: Text Body: Text Header: Text

Contexts Default Context Sites

Fuzzer 13.0.1 Advanced fuzzer for manual testing

Getting Started with ZAP Guide 11.0.0 A short Getting Started with ZAP Guide

Help - English 10.0.0 English version of the ZAP help file.

HUD - Heads Up Display 0.12.0 Display information from ZAP in browser.

Import files containing URLs 7.0.0 Adds an option to import a file of URLs. The file must be plain text with...

Invoke Applications 10.0.0 Invoke external applications passing context related information such...

Online menus 7.0.0 ZAP Online menu items

OpenAPI Support 16.0.0 Imports and spiders OpenAPI definitions.

Passive scanner rules 29.0.0 The release quality Passive Scanner rules

Quick Start 28.0.0 Provides a tab which allows you to quickly test a target application

Replacer 8.0.0 Easy way to replace strings in requests and responses.

Reveal 3.0.0 Show hidden fields and enable disabled fields

Save Raw Message 5.0.0 Allows to save content of HTTP messages as binary

Status Beta Version 13.0.1 Description Advanced fuzzer for manual testing

Changes Fixed

Fix exception when saving the options with no default category selected (Issue 6136).

Info <https://www.zaproxy.org/docs/desktop/addons/fuzzer/>

Repo <https://github.com/zaproxy/zap-extensions/>

Id fuzz

Author ZAP Dev Team

Not Before Version 2.9.0

File C:\Users\renders\OWASP ZAP\plugin\fuzz-beta-13.0.1.zap

Uninstall Selected Update Selected Up

Filter: OFF Export

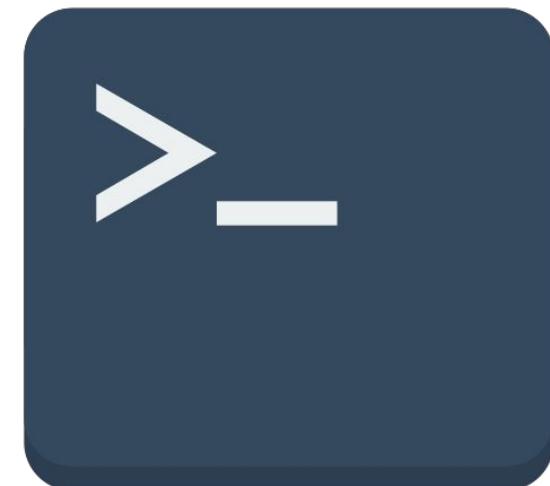
Id	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body

Updated download, see Output tab for details

ZAP for DAST



Automating DAST



Deploying ZAP



Learn more!

ZAP Resources

- <https://www.zaproxy.org/>
- <https://www.alldaydevops.com/zap-in-ten>
- <https://github.com/Grunny/zap-cli>

OWASP

- <https://owasp.org>

OWASP Toronto

- <https://owasp.org/www-chapter-toronto/>
- <https://meetup.com/OWASP-Toronto/>

OWASP ASVS

A brief overview

Intro to OWASP Application Security Verification Standard (ASVS)

OWASP Flagship Project!



Open source!
Creative Commons Attribution ShareAlike 3.0 license.



Meant to be a common framework of application security verification requirements

- Normalizing range of coverage
- Normalizing level of rigour



ASVS version 4.0.2

Applicability		Building			Building, Configuration, Deployment Assurance and Verification			Assurance and Verification	
Level 1	All apps		Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Penetration Testing	DAST
Level 2	All apps	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Level 3	High Assurance	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Legend		Acceptable	Suitable						

Levels

ASVS Level 1

- for low assurance levels, and is completely “penetration testable”

ASVS Level 2

- for applications that contain sensitive data which requires protection and is the recommended level for most apps

ASVS Level 3

- for the most critical applications - applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

Example content:

V3.2 Session Binding Requirements

#	Description	L1	L2	L3	CWE	NIST §
3.2.1	Verify the application generates a new session token on user authentication. (C6)	✓	✓	✓	384	7.1
3.2.2	Verify that session tokens possess at least 64 bits of entropy. (C6)	✓	✓	✓	331	7.1
3.2.3	Verify the application only stores session tokens in the browser using secure methods such as appropriately secured cookies (see section 3.4) or HTML 5 session storage.	✓	✓	✓	539	7.1
3.2.4	Verify that session token are generated using approved cryptographic algorithms. (C6)	✓	✓	331	7.1	

Example content:

V5.1 Input Validation Requirements

Properly implemented input validation controls, using positive allow lists and strong data typing, can eliminate more than 90% of all injection attacks. Length and range checks can reduce this further. Building in secure input validation is required during application architecture, design sprints, coding, and unit and integration testing. Although many of these items cannot be found in penetration tests, the results of not implementing them are usually found in V5.3 - Output encoding and Injection Prevention Requirements. Developers and secure code reviewers are recommended to treat this section as if L1 is required for all items to prevent injections.

#	Description	L1	L2	L3	CWE
5.1.1	Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables).	✓	✓	✓	235
5.1.2	Verify that frameworks protect against mass parameter assignment attacks, or that the application has countermeasures to protect against unsafe parameter assignment, such as marking fields private or similar. (C5)	✓	✓	✓	915
5.1.3	Verify that all input (HTML form fields, REST requests, URL parameters, HTTP headers, cookies, batch files, RSS feeds, etc) is validated using positive validation (allow lists). (C5)	✓	✓	✓	20
5.1.4	Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers or telephone, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match). (C5)	✓	✓	✓	20
5.1.5	Verify that URL redirects and forwards only allow destinations which appear on an allow list, or show a warning when redirecting to potentially untrusted content.	✓	✓	✓	601

Uses

As a metric for verifications and assessments

As input for secure development training

As detailed security architecture guidance

As a driver for agile application security

As secure coding checklist

As a framework for guiding software procurement

As guide for automated testing

To find out more ...

ASVS project page

- <https://owasp.org/www-project-application-security-verification-standard/>

OWASP Toronto

- <https://owasp.org/www-chapter-toronto/>
- <https://meetup.com/OWASP-Toronto/>

OWASP

- <https://owasp.org>



Life is too short • AppSec is tough • Cheat!

<https://cheatsheetseries.owasp.org/>

Cross Site Scripting Prevention

Search

RULE #0 - Never Insert Untrusted Data Except in Allowed Locations

The first rule is to **deny all** - don't put untrusted data into your HTML document unless it is within one of the slots defined in Rule #1 through Rule #5. The reason for Rule #0 is that there are so many strange contexts within HTML that the list of encoding rules gets very complicated. We can't think of any good reason to put untrusted data in these contexts. This includes "nested contexts" like a URL inside a JavaScript -- the encoding rules for those locations are tricky and dangerous.

If you insist on putting untrusted data into nested contexts, please do a lot of cross-browser testing and let us know what you find out.

Directly in a script:

```
<script>...NEVER PUT UNTRUSTED DATA HERE...</script>
```

Index Alphabetical

65 cheat sheets available.

Icons beside the cheat sheet name indicate in which language(s) code snippet(s) are provided.

A B C D E F G H I J K L M N O P Q R S T U V W X

A

AJAX Security Cheat Sheet. (JSON)
Access Control Cheat Sheet.
Authentication Cheat Sheet.
Abuse Case Cheat Sheet.
Attack Surface Analysis Cheat Sheet.
Authorization Testing Automation Cheat

Index ASVS

Search

OWASP/CheatSheetSeries
13.9k Stars · 2k Forks

Objective

The objective of this index is to help an OWASP Application Security Verification Standard (ASVS) user clearly identify which cheat sheets are useful for each section during his or her usage of the ASVS.

This index is based on the version 4.x of the ASVS.

V1: Architecture, Design and Threat Modeling Requirements

V1.1 Secure Software Development Lifecycle Requirements

Threat Modeling Cheat Sheet.

Abuse Case Cheat Sheet.

Attack Surface Analysis Cheat Sheet

V1.2 Authentication Architectures

None.

V1.3 Session Management

Table of contents

A
B
C
D
E
F
H

Table of contents

Table of Contents
Objective
V1: Architecture, Design and Threat Modeling Requirements
V1.1 Secure Software Development Lifecycle Requirements
V1.2 Authentication Architectural Requirements
V1.3 Session Management Architectural Requirements
V1.4 Access Control Architectural Requirements

Index Proactive Controls

Search

OWASP/CheatSheetSeries
13.9k Stars · 2k Forks

Objective

This cheatsheet will help users of the OWASP Proactive Controls identify which cheatsheets map to each proactive controls item. This mapping is based the OWASP Proactive Controls version 3.0 (2018).

1. Define Security Requirements

Abuse Case Cheat Sheet

Attack Surface Analysis Cheat Sheet

Threat Modeling Cheat Sheet

2. Leverage Security Frameworks and Libraries

Clickjacking Defense Cheat Sheet

DotNet Security Cheat Sheet (A3 Cross Site Scripting)

Table of contents

Table of Contents
Objective
1. Define Security Requirements
2. Leverage Security Frameworks and Libraries
3. Secure Database Access
4. Encode and Escape Data
5. Validate All Inputs
6. Implement Digital Identity
7. Enforce Access Controls
8. Protect Data Everywhere
9. Implement Security Logging and Monitoring
10. Handle All Errors and Exceptions

OWASP SAMM

(Software Assurance Maturity Model)

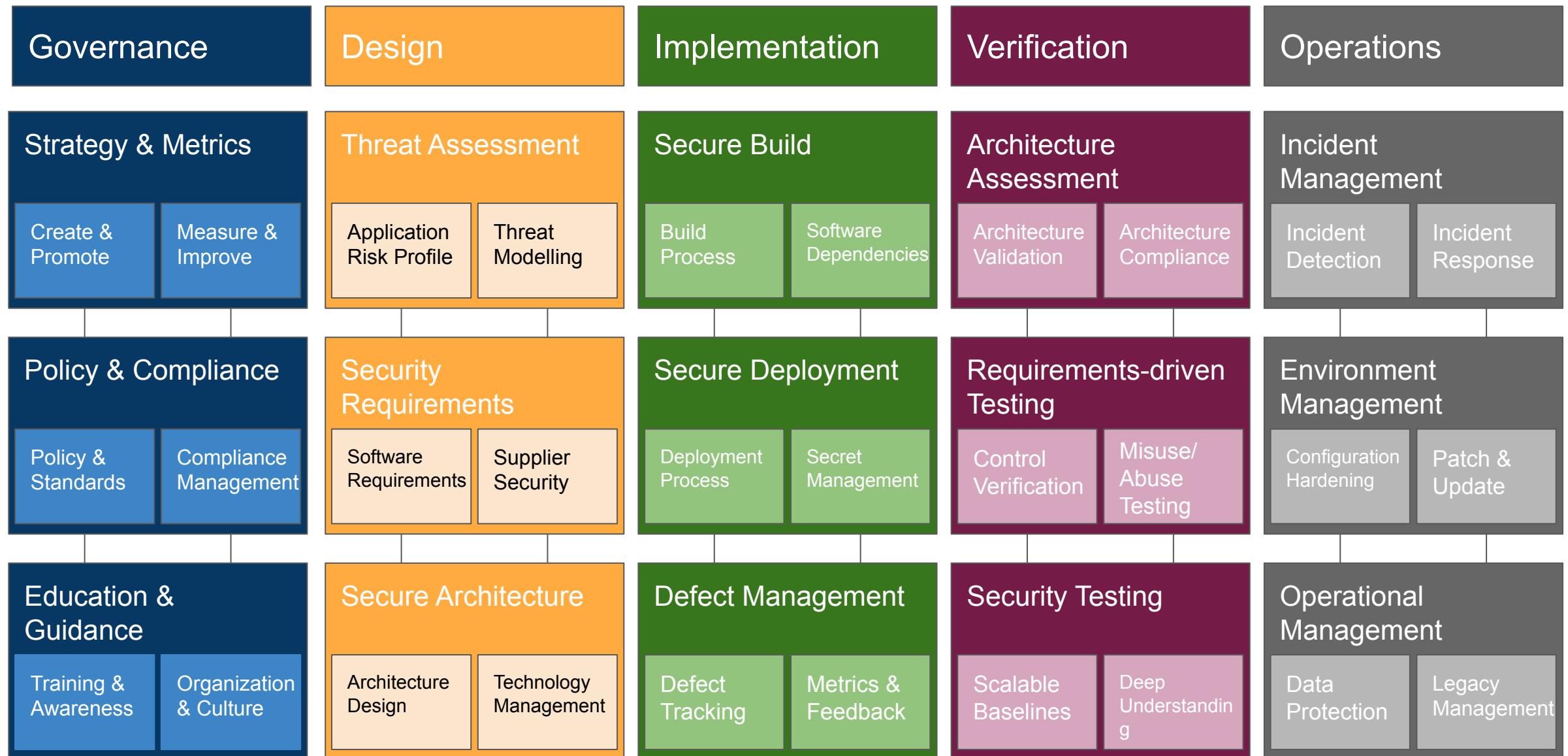
Where to find it?

Main Site: <https://owaspSAMM.org/>

Guidance:

- <https://owaspSAMM.org/guidance/quick-start-guide/>
- <https://owaspSAMM.org/guidance/agile/>
- <https://github.com/OWASP/samm/tree/master/Supporting%20Resources/v2.0/toolbox>
- <https://concordusa.com/SAMM/>

Model Overview





Thank you!

Yuk Fai Chan, yukfai.chan@owasp.org

Adam Greenhill, adam.greenhill@owasp.org

Jack Enders, jack.enders@owasp.org

Opheliar Chan, opheliar.chan@owasp.org

[https://www.meetup.com/
OWASP-Toronto/](https://www.meetup.com/OWASP-Toronto/)



Other Projects

```

,+++
:+
+|@#####
&@# + .o@##.
&@& .@#@#W, o@## :@#&W&@ ,@# .: o@+ ,@#+++&#&
+@& &@& #@& +@#&@#& :@V, +@& +@: .@#
S@ @@ S@o S@S MM ,@V W@+ @V, o@#;
VV S@o S@: o@+ o@+ #@. S@o +W@#+, +W@S:
#@ :@V S@+ S@+ o@: @o o@o o@o@U+ o@S
o@+ @@& S@+ S@+ #@ S@. ,W@V +#@& o@V.
VV +@V@S, S@+ ;@ o@+ #@ :@M@& S@: .. ;@o
:@W: o@# +W@ S@+ ;@: +@W&o++o@W. S@& S@o+@V. #@: o@+
:W@WVW@#S + :S@W@o@& &V .o@o@V&. :W@W@o@&
+o@&&&+, +0000.

```

v3.5.3
 OWASP Amass Project - @owaspamass
 In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel|enum|viz|track|db|dns [options]

- h Show the program usage message
- help Show the program usage message
- version Print the version number of this Amass binary

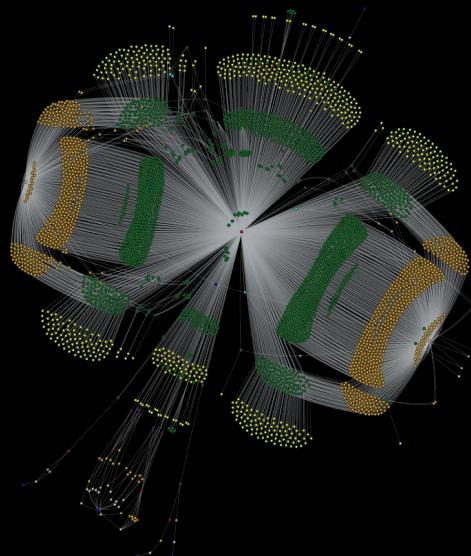
Subcommands:

- amass intel - Discover targets for enumerations
- amass enum - Perform enumerations and network mapping
- amass viz - Visualize enumeration results
- amass track - Track differences between enumerations
- amass db - Manipulate the Amass graph database
- amass dns - Resolve DNS names at high performance

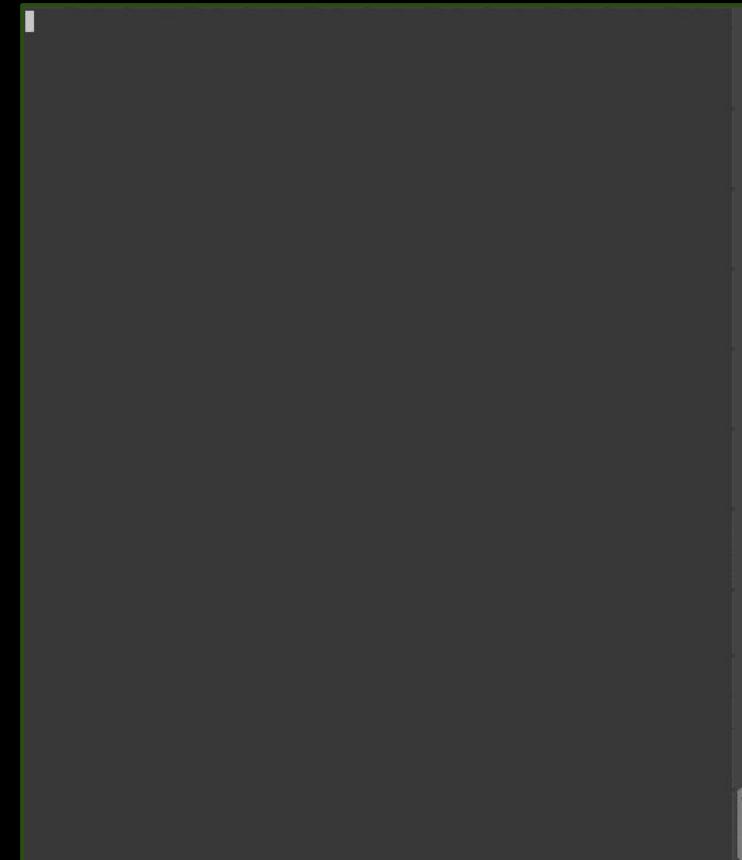
The user's guide can be found here:
https://github.com/OWASP/Amass/blob/master/doc/user_guide.md

An example configuration file can be found here:
<https://github.com/OWASP/Amass/blob/master/examples/config.ini>

The Amass tutorial can be found here:
<https://github.com/OWASP/Amass/blob/master/doc/tutorial.md>



<https://owasp.org/www-project-amass/>



DefectDojo



DEFECTDOJO

Bodgeit F vulnerable

Overview Metrics Engagements 38 Findings 204 Endpoints 27 Benchmarks Settings

Description

Bodgeit

- Easy to install - just requires java and a servlet engine, e.g. Tomcat
- Self contained (no additional dependencies other than to 2 in the above line)
- Easy to change on the fly - all the functionality is implemented in JSPs, so no IDE required* Cross platform
- Open source* No separate db to install and configure - it uses an 'in memory' db that is automatically (re)initialized on start up
- More testing

Metrics

CRITICAL	33	80	85	6	204
HIGH	MEDIUM	LOW	INFORMATIONAL	TOTAL	

Technologies Apache 2.0 v.1

Regulations (1) GDPR EU & EU Data Extra-Territorial Applicability Privacy

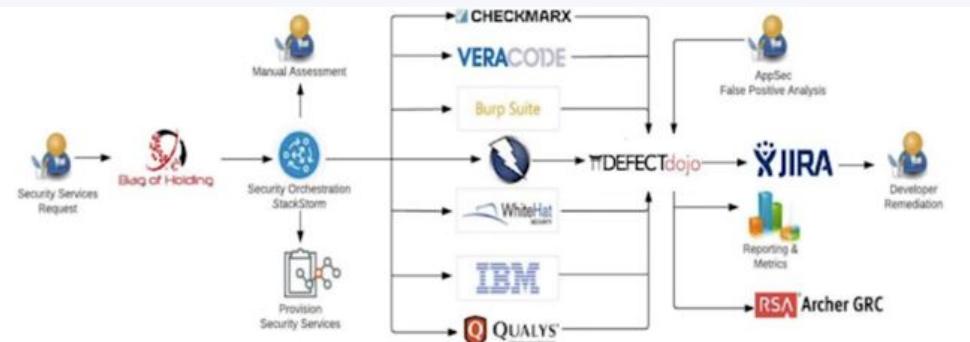
Metadata

Business Criticality	High
Product Type	Research and Development
Platform	Web
Lifecycle	Construction
Origin	Third Party Library
User Records	1,000
Revenue	50,000.00

Languages

Total Files: 55 Lines of Code: 10,400

<https://owasp.org/www-project-defectdojo/>
<https://github.com/DefectDojo/>



DefectDojo latest

Search docs

About DefectDojo Getting Started

Integrations

- Acunetix Scanner
- Anchore-Engine
- Aqua
- Arachni Scanner
- AppSpider (Rapid7)
- AWS Security Hub
- AWS Scout2 Scanner
- AWS Prowler Scanner
- Bandit
- Blackduck Hub
- Brakeman Scan
- Bugcrowd
- Bundler-Audit
- Burp XML
- Burp Enterprise Scan
- CCVS Report
- Checkow Report

Read the Docs v. latest

Docs » Integrations

Integrations

DefectDojo has the ability to import reports from other security tools.

Acunetix Scanner

XML format.

Anchore-Engine

JSON vulnerability report generated by anchore-cli tool, using a command like `anchore-cli --json image vuln <image>:tag all`

Aqua

JSON report format.

Arachni Scanner

Arachni JSON report format.

AppSpider (Rapid7)

Use the VulnerabilitiesSummary.xml file found in the zipped report download.

Burp-Plugin

A Burp plugin to export findings to DefectDojo

Installation

In order for the plugin to work, you will need to have Jython set up in Burp Suite Pro. To use this plugin before it appears in the App Store you will need to do the following:

1. Go to `Extender` and select the `Extensions` tab
2. Click on `Add`, select `Extension Type` to be `Python` and select the `DefectDojoPlugin.py`

Usage

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Composer Extender Project options User options RON Web Takers Hackinator Help DefectDojo

DefectDojo: defectdojohost:443 Username: admin API key: 1edf9f2475b7ec7519c16f6910b Product Name: Budget

Product ID: 1 Service Host: Search_P Engagement ID: 1|1|1|Editor Impact: Fis 17-Aug-2018 18:00:55 Test ID: 2|2|Burp Scan 2018-09-17T16:29:55.91900|Send issue|

http://www.googleapis.com Cleartext traffic connection accepted

http://www.whatsnew.googleapis.com Cleartext subresource connection accepted

http://defectdojohost:443 Burp Scan 2018-09-17T16:29:55.91900|Send issue|

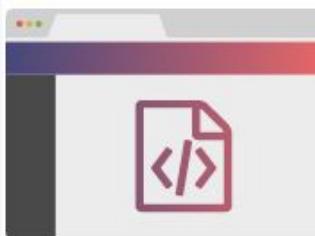
http://www.googleapis.com|Cleartext traffic connection accepted|Cleartext subresource connection accepted|Password field with autocomplete enabled|Cross domain referrer leakage|Cross domain cookie leakage|



Training developers in writing secure code

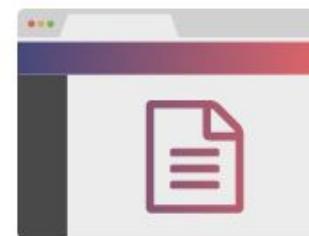
SKF is a fully open-source Python-Flask web-application that uses the OWASP Application Security Verification Standard to train you and your team in writing secure code, by design.

<https://securityknowledgeframework.org>



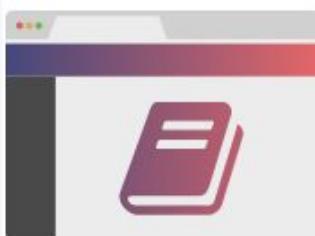
Detect possible threats in your application

In pre-development detect possible threats based on the processing functions on your application.



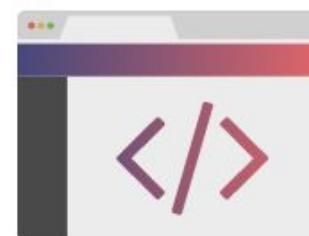
Run OWASP ASVS Checklists

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.

Checklist options

Checklist **Description** **Active** **Show list**

Architecture, Design and Threat Modeling Requirements	Security architecture has almost become a lost art in many organizations. The days of the enterprise architect have passed in the age of DevSecOps. The application security field must catch up and adopt agile security principles while re-introducing leading security architecture principles to software practitioners. Architecture is not an implementation, but a way of thinking about a problem that has potentially many different answers, and no one single "correct" answer. All too often, security is seen as inflexible and demanding that developers fix code in a particular way, when the developers may know a much better way to solve the problem. There is no single, simple solution for architecture, and to pretend otherwise is a disservice to the software engineering field.	True	
Authentication Verification Requirements	Authentication is the act of establishing, or confirming, someone (or something) as authentic and that claims made by a person or about a device are correct, resistant to impersonation, and prevent recovery or interception of passwords.	True	

Checklist options

Knowledge Base

Code examples

Labs

Supported by OWASP

X XSS Protection Header: asp

XXSSProtection header

Example:

```
/*
In order to set the XXSSProtection header, you'll have to add the following
code to the head of your application, the following code could be used in your controller:
*/
```

Response.AppendHeader("XXSSProtection:1", "mode=block");

```
/*
In your classes you can use the following code:
*/
```

HttpContext.Current.Response.AppendHeader("XXSSProtection:1", "mode=block");

Checklist options

Knowledge Base

In depth information on how to approach specific functionality or problems with explanation of attack surface and mitigations!

Add item

Web applications

Search vulnerability

- [Filename injection Path traversal](#)
- [XSS injection](#)
- [Command injection](#)

Checklist options

Knowledge Base

Code examples

Labs

Supported by OWASP

Search lab

Lab Name	Skill level	To Lab
Path traversal (LFI)	1	
Cross Site Scripting	1	
Cross site scripting (attribute)	1	

<https://demo.securityknowledgeframework.org/dashboard>



OWASP ModSecurity Core Rule Set

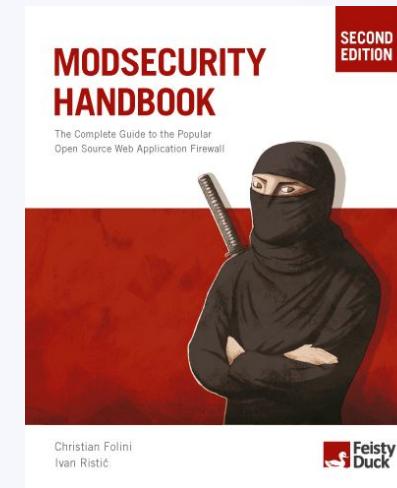
THE 1ST LINE OF DEFENSE

<https://coreruleset.org/>

```
# --=[ HTTP Splitting ]=-
#
# This rule detect \n or \r in the REQUEST FILENAME
# Reference: https://www.owasp.org/index.php/Testing_for_HTTP_Splitting/Smuggling_(OTG-INPVAL-016)
#
SecRule REQUEST_FILENAME "@rx [\\n\\r]" \
    "id:921190, \
    phase:1, \
    block, \
    t:none,t:urlDecodeUni, \
    msg:'HTTP Splitting (CR/LF in request filename detected)', \
    logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}: %{MATCHED_VAR}', \
    tag:'application-multi', \
    tag:'language-multi', \
    tag:'platform-multi', \
    tag:'attack-protocol', \
    tag:'paranoia-level/1', \
    tag:'OWASP_CRS', \
    tag:'capec/1000/210/272/220/34', \
    ctl:auditLogParts=+E, \
    ver:'OWASP_CRS/3.3.0', \
    severity:'CRITICAL', \
    setvar:'tx.httpViolationScore+=%{tx.criticalAnomalyScore}', \
    setvar:'tx.anomalyScore_P1+=%{tx.criticalAnomalyScore}'"
```



<https://youtu.be/5qW9IUNLGqQ>



<https://www.feistyduck.com/library/modsecurity-handbook-free/online/>