# 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!

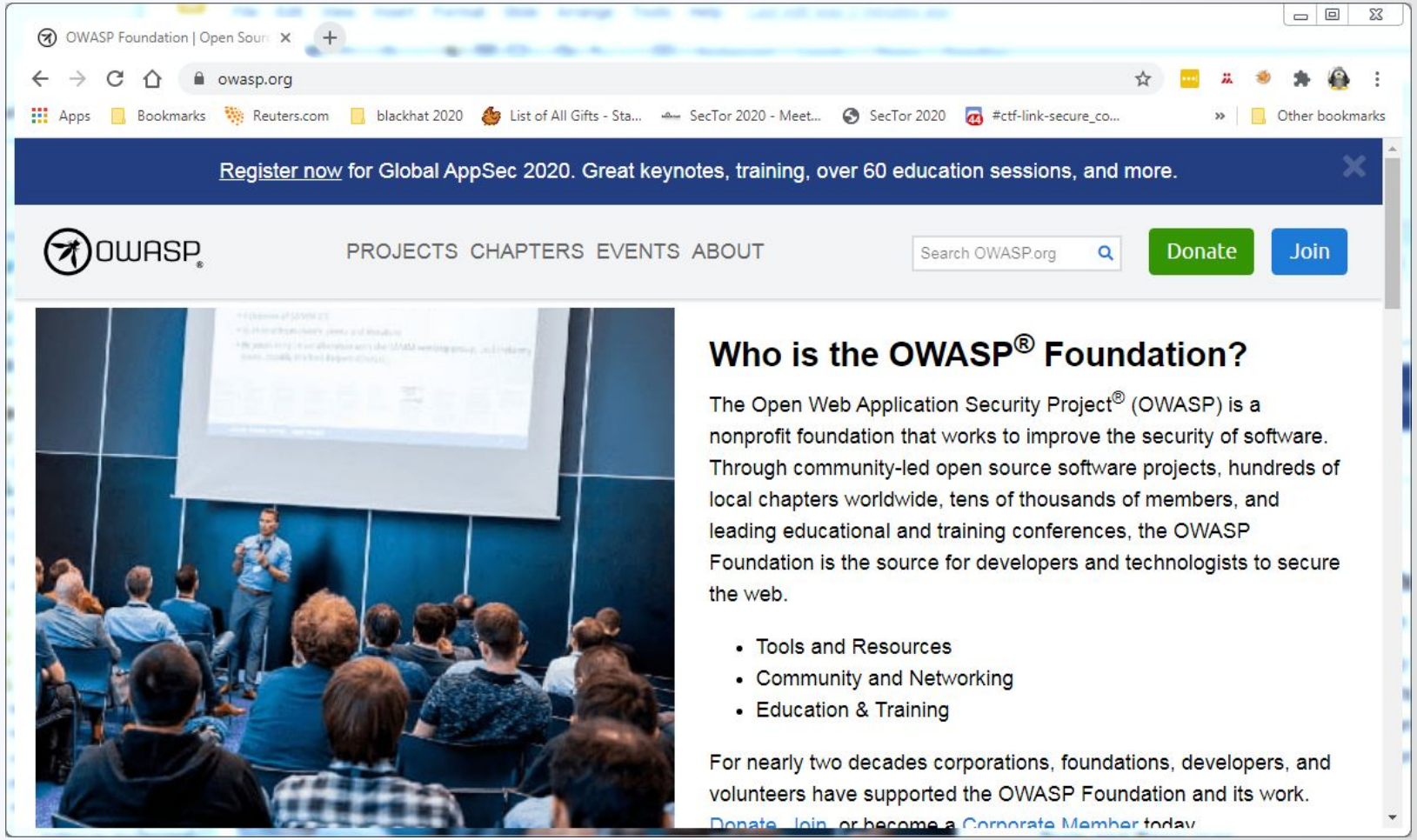

**45,000+ OWASP volunteers worldwide**

# World Wide

- 207 local Chapters in 56 countries… and counting!

# OWASP.ORG

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



# OWASP Projects

- **189 Projects including 20 Flagship Projects**

### Flagship Projects

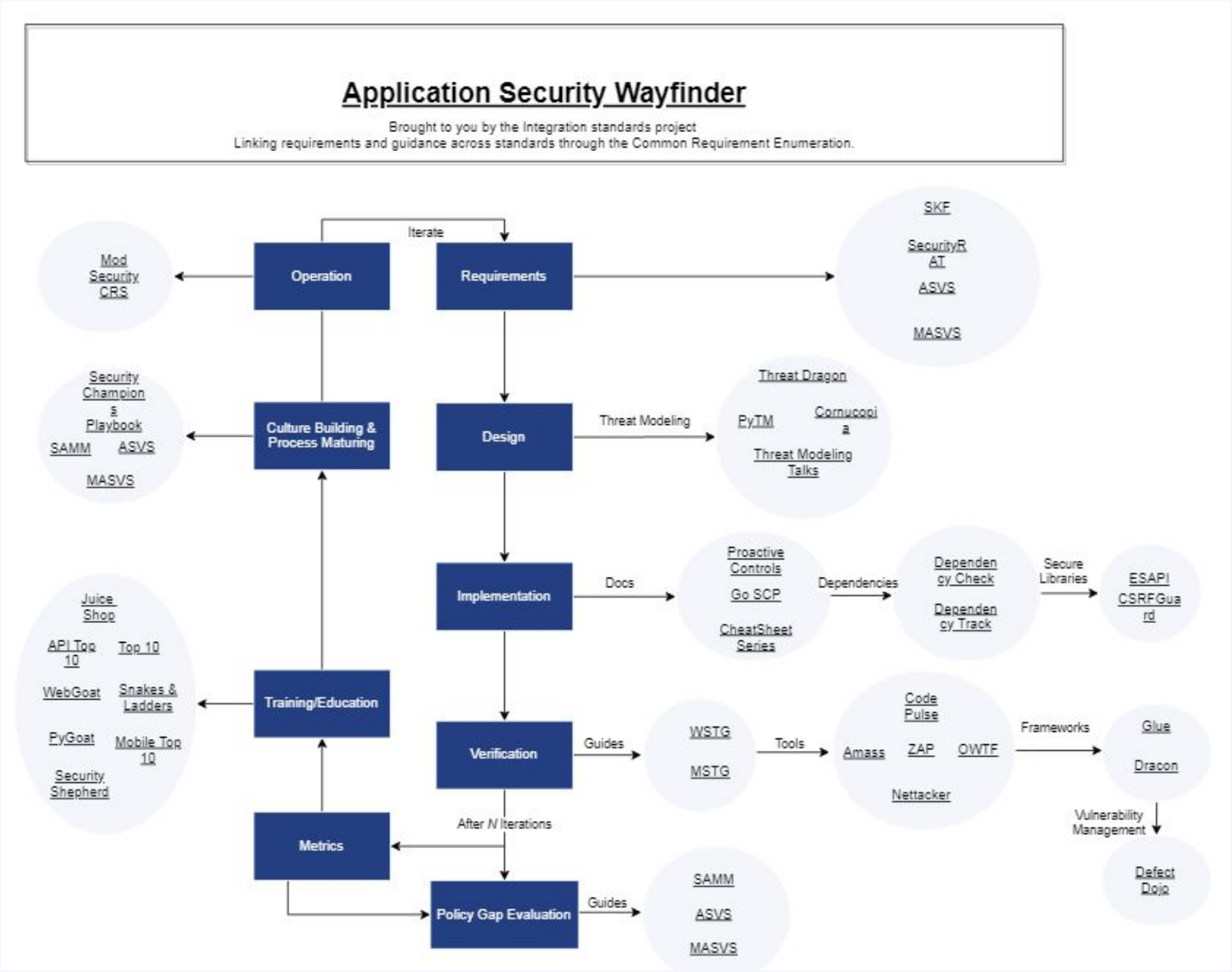
- OWASP Amass
- OWASP Application Security Verification Standard
- OWASP Cheat Sheet Series
- OWASP CSRFGuard
- OWASP Defectdojo
- OWASP Dependency-Check
- OWASP Dependency-Track
- OWASP Juice Shop
- OWASP Maryam
- OWASP Mobile Security Testing Guide
- OWASP Mobile Top 10
- OWASP ModSecurity Core Rule 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/




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/

# Its 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):


## New Corporate Members

## 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

**@OWASP**

"**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)

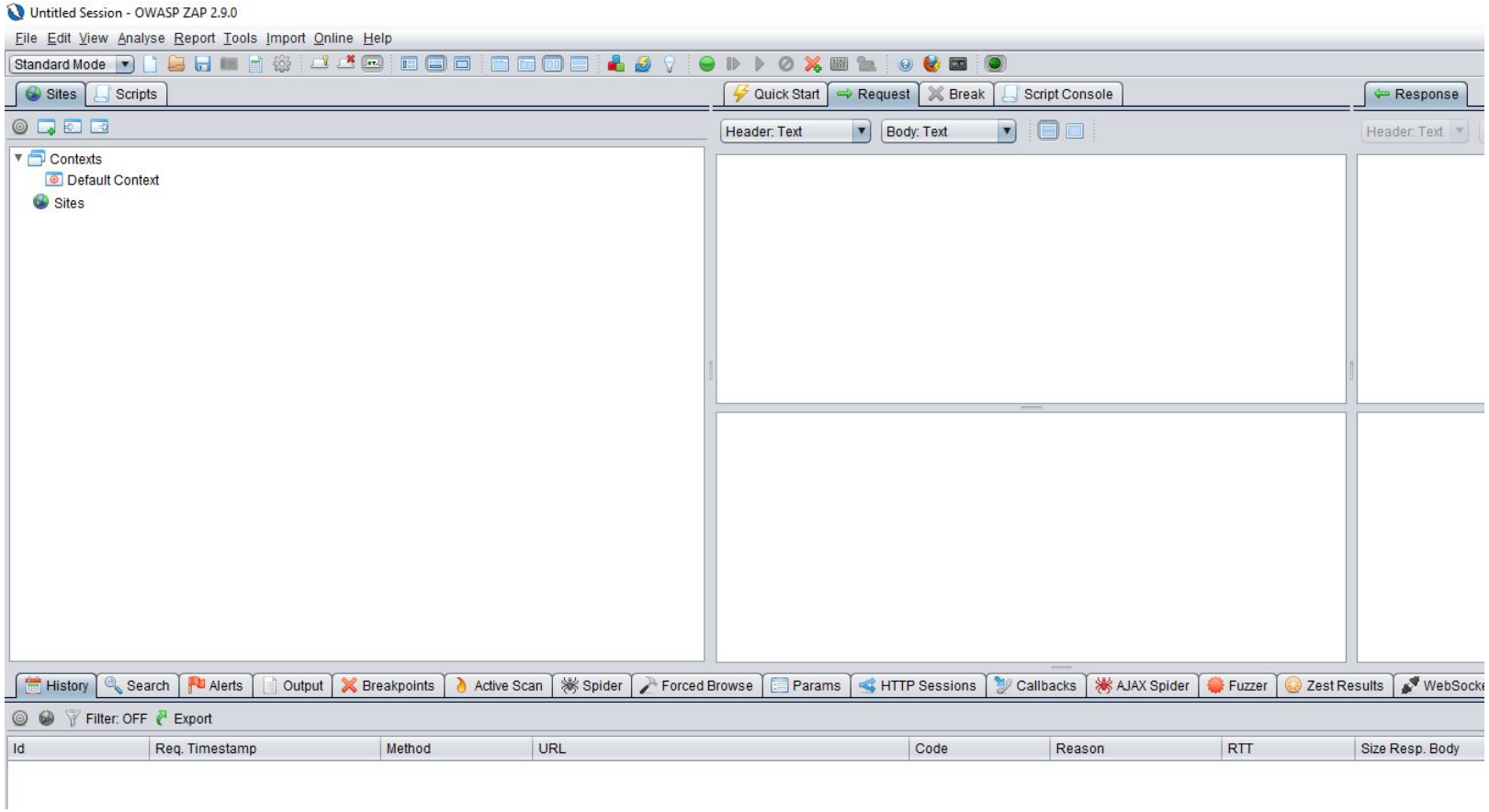
OWASP Flagship Project!

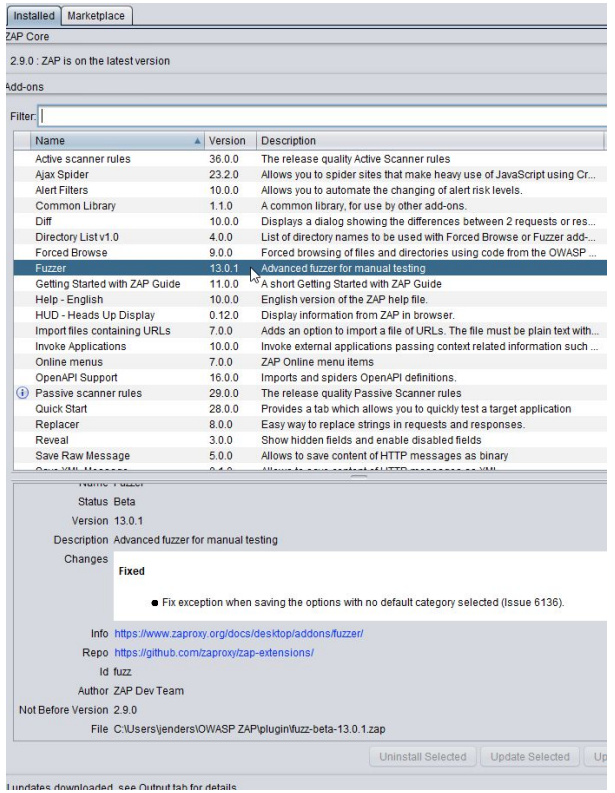
Free and Open source!

*Apache 2 License*

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

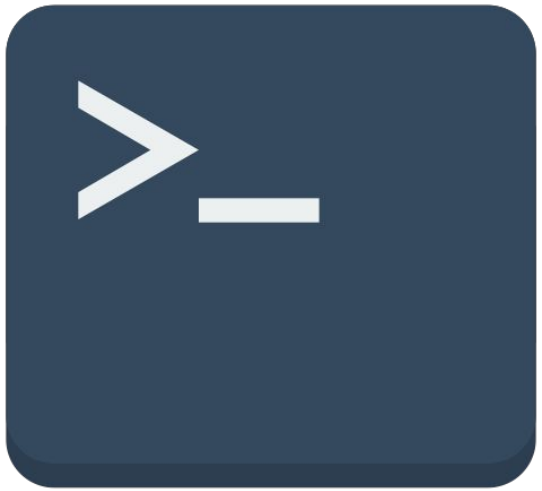

# ZAP for Testers



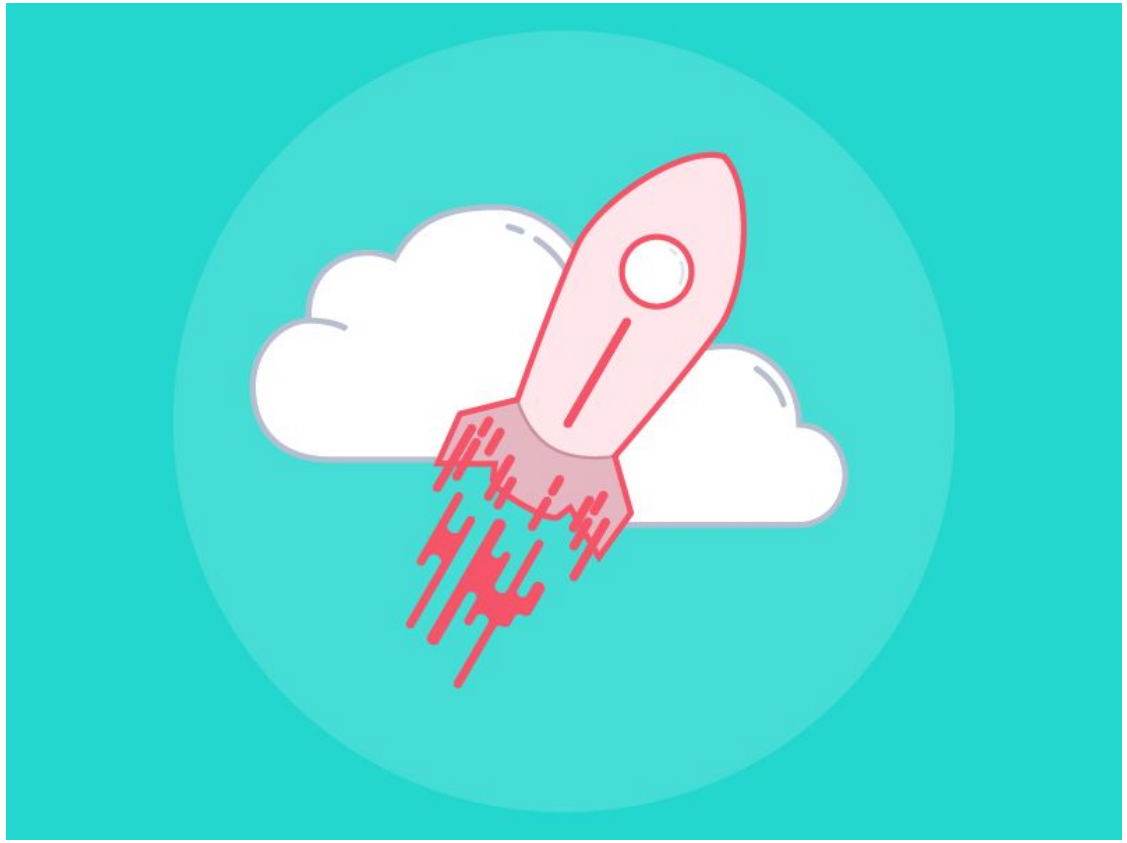



# ZAP for DAST

# 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!**

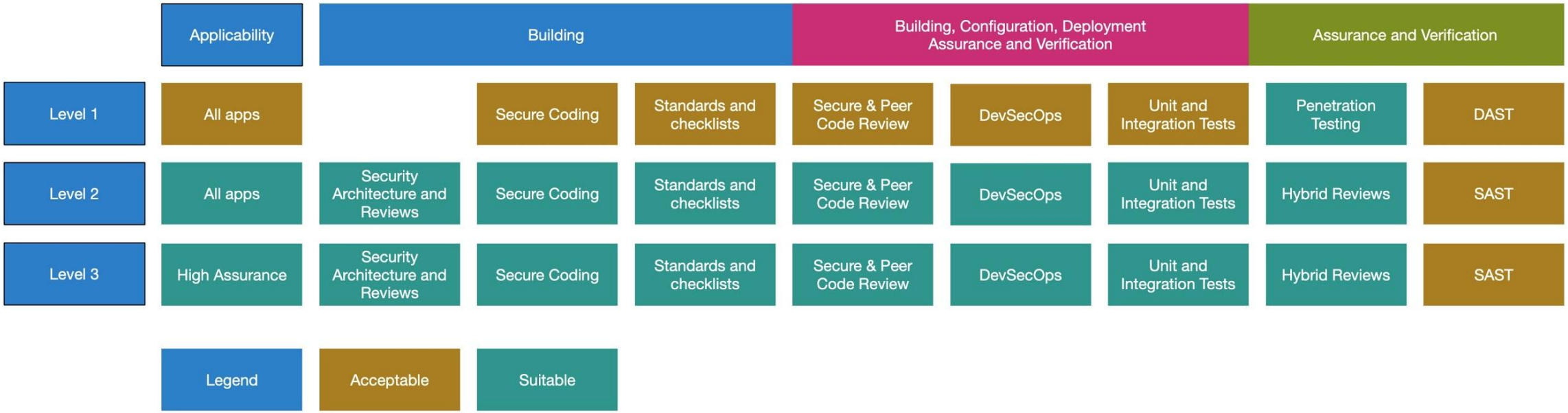
**Open source!**

*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



# 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 detailed security architecture guidance

As secure coding checklist

As guide for automated testing

As input for secure development training

As a driver for agile application security

As a framework for guiding software procurement

# To find out more ...

ASVS project page

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

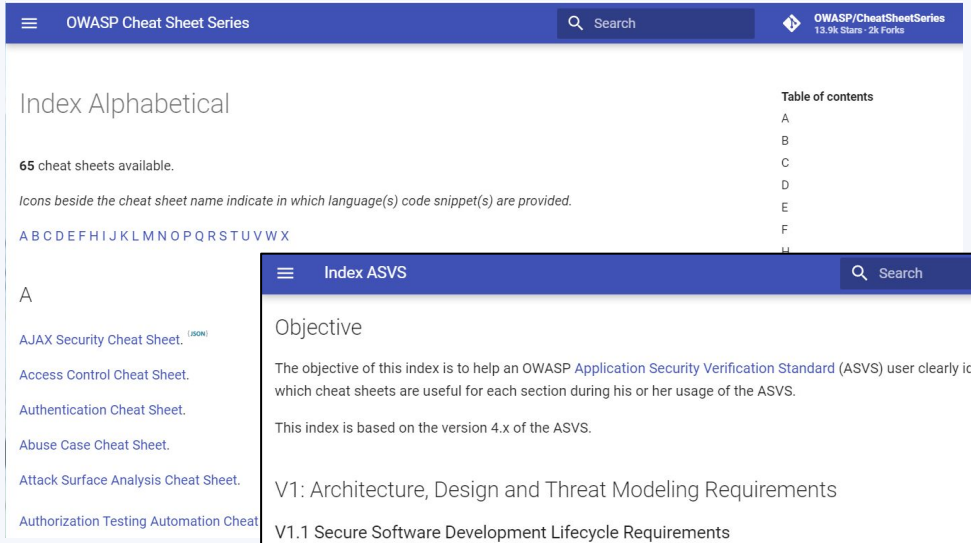
OWASP Toronto

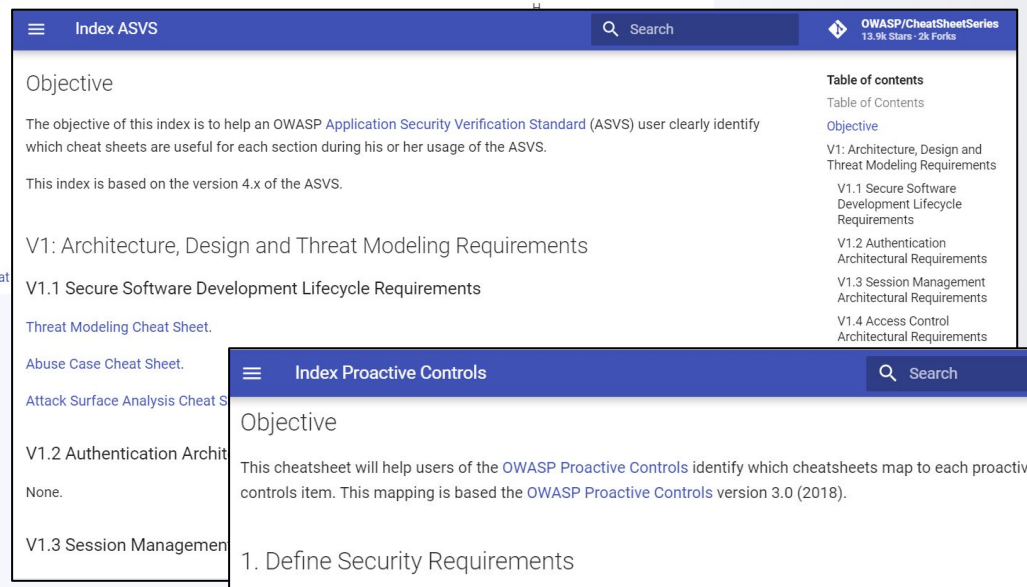
- https://owasp.org/www-chapter-toronto/
- https://meetup.com/OWASP-Toronto/

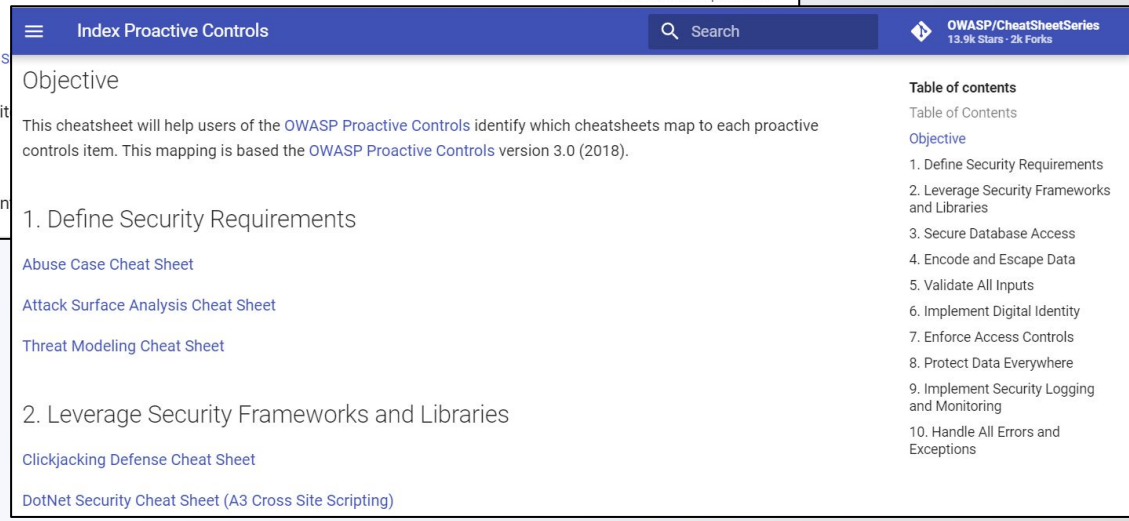
OWASP

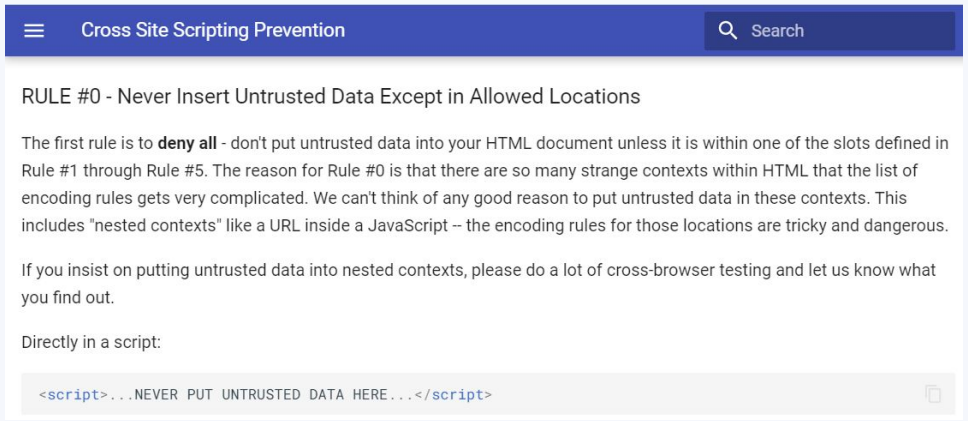
- https://owasp.org


https://cheatsheetsseries.owasp.org/




**IMPROVING SOFTWARE SECURITY, WORLDWIDE**

# OWASP SAMM

**(Software Assurance Maturity Model)**

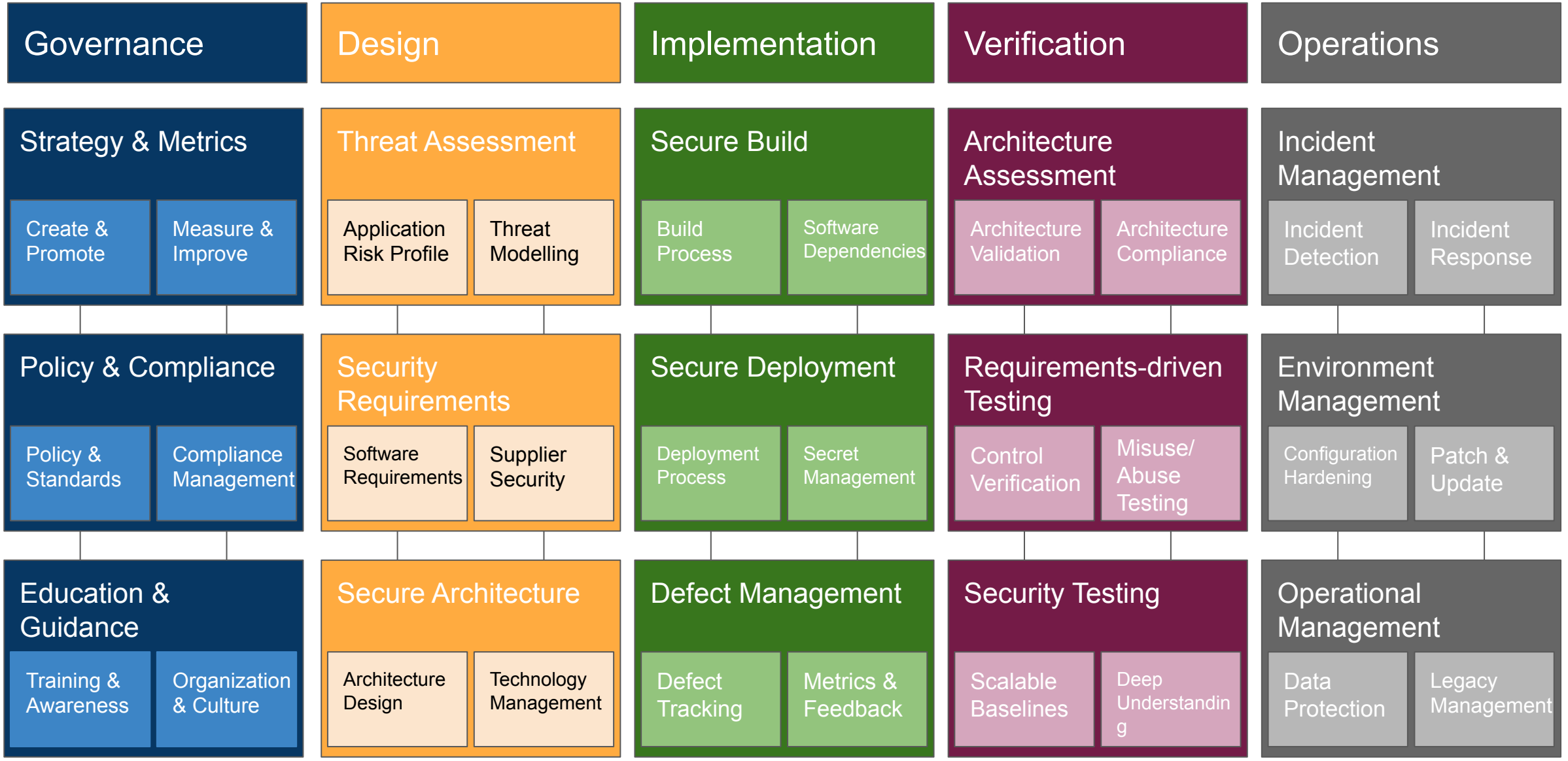
# Where to find it?

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



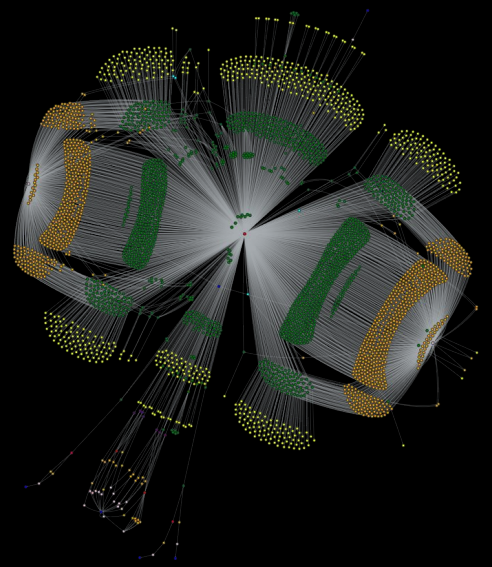
# Thank you!

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/

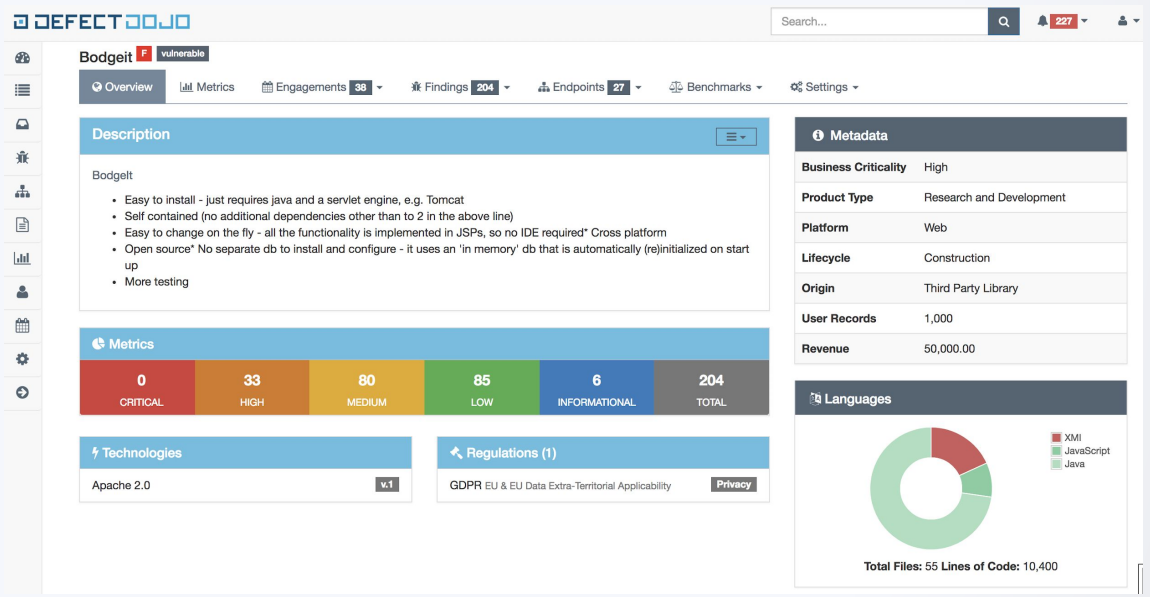
# Other Projects

```
        .+++:.            :                             .+++.
      +W@@@@@@8        &+W@#               o8W8:      +W@@@@@@#.   oW@@@W#+
     &@#+    .o@##.    .@@@o@W.o@@o       :@@#&W8o    .@#:  .:oW+  .@#+++&#&
    +@&        &@&     #@8 +@W@&8@+     :@W.    +@8   +@:          .@8
    8@          @@     8@o  8@8  WW    .@W      W@+  .@W.          o@#:
    WW          &@o    &@:  o@+  o@+   #@.      8@o   +W@#+.         +W@8:
    #@          :@W    &@+  &@+   @8  :@o       o@o     oW@@W+        oW@8
    o@+          @@&   &@+  &@+   #@  &@.      .W@W       .+#@&         o@W.
     WW         +@W@8. &@+  :&    o@+ #@      :@W&@&                &@:  ,,     :@o
     :@W:      o@# +Wo &@+        :W:  +@W&o++o@W. &@&  8@#o+&@W.  #@:    o@+
      :W@@WWWW@@8       +             :&W@@@@&    &W  .o#@@W&.   :W@WWW@@&
        +o&&&&+.                                                    +oooo.

                                                                    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
```

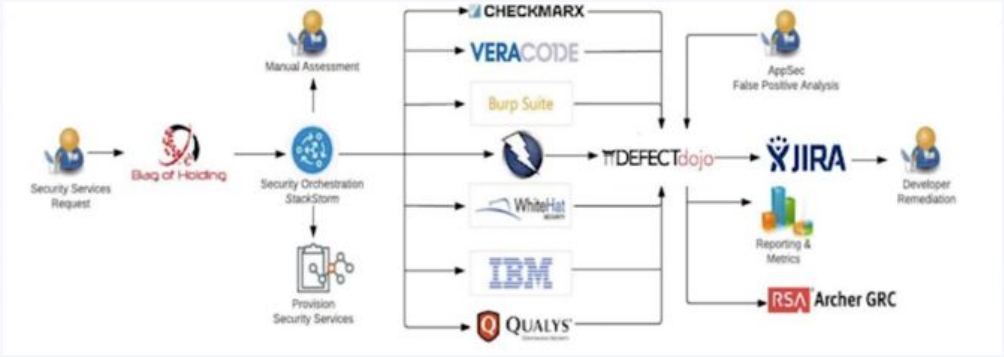


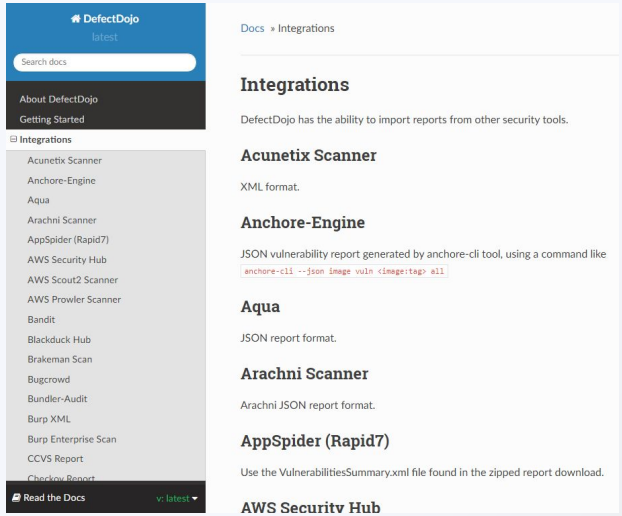
https://owasp.org/www-project-amass/

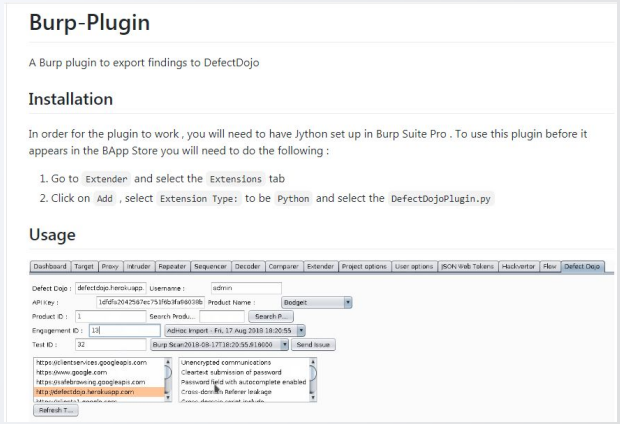
# DefectDojo




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



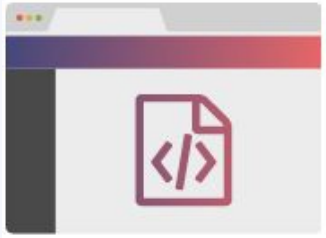

## IMPROVING SOFTWARE SECURITY, WORLDWIDE

Security Knowledge Framework

## 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.or



### Detect possible threats in your application

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

### Run OWASP ASVS Checklists

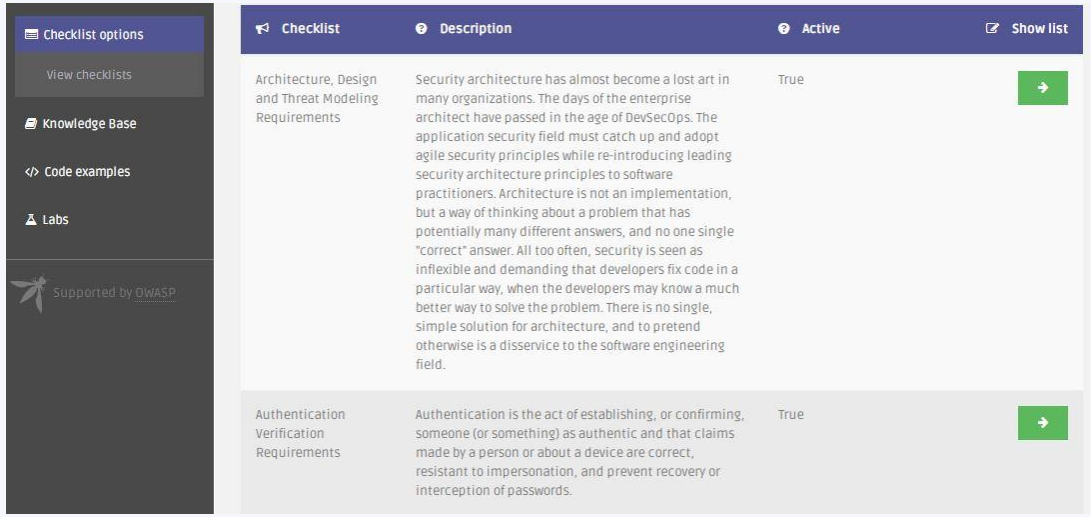
Harden your application functions in post-development by running OWASP ASVS checklists, complete with feedback and solutions.

### Learn about threats and vulnerabilities in the SKF knowledge base

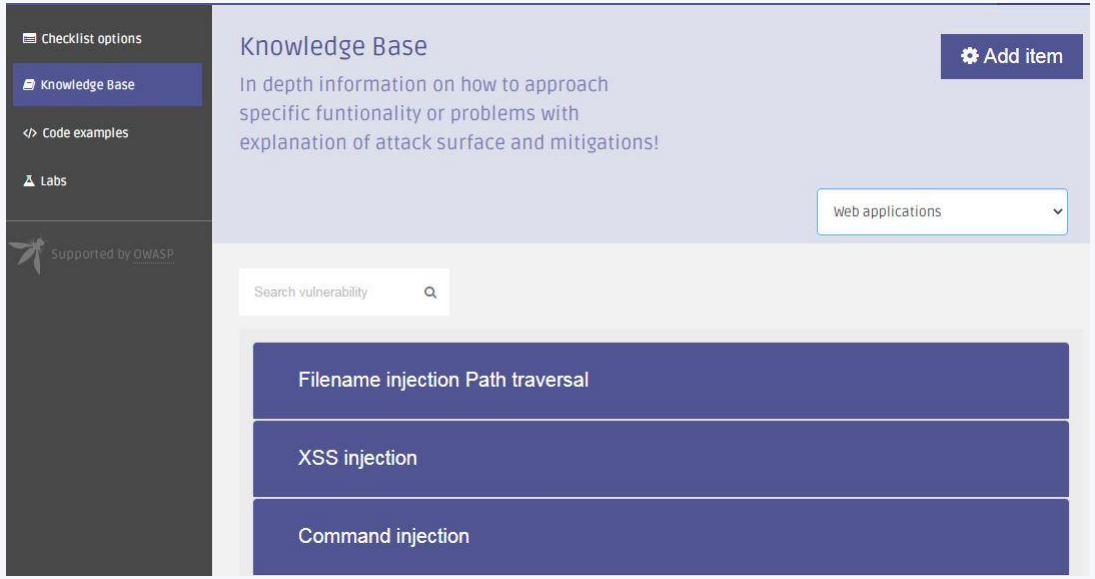
An extensive library of common hacks and exploits, learn the hacker mindset and keep your project secure.

### Learn to code secure from best practice code examples

An extensive library of code examples for a wide range of functions, beautifully commented.









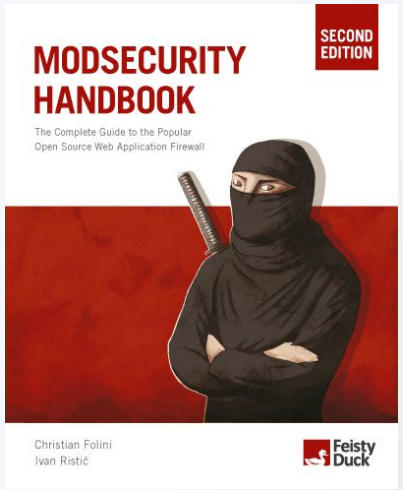
https://demo.securityknowledgeframework.org/dashboard

OWASP
ModSecurity
Core Rule Set
THE 1ST LINE OF DEFENSE

https://coreruleset.org/


https://youtu.be/5qW9IUNLGqQ

```
# -=[ 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.http_violation_score=+%{tx.critical_anomaly_score}',\
    setvar:'tx.anomaly_score_pl1=+%{tx.critical_anomaly_score}'"
```




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