Overview of OWASP & Intro to OWASP Projects

OWASP Toronto

March 2017

Who is OWASP?

Software powers the world, but insecure software threatens safety, trust, and economic growth. The Open Web Application Security Project (OWASP) is dedicated to making application security visible by empowering individuals and organizations to make informed decisions about true application security risks.

OWASP's DNA

The value of volunteerism

- 94% think volunteering adds to the skills of their workforce
- 58% say voluntary work can be more valuable than experience gained in paid employment
- 25% offer paid time off to employee volunteers
- 15% allow sabbaticals for volunteering projects

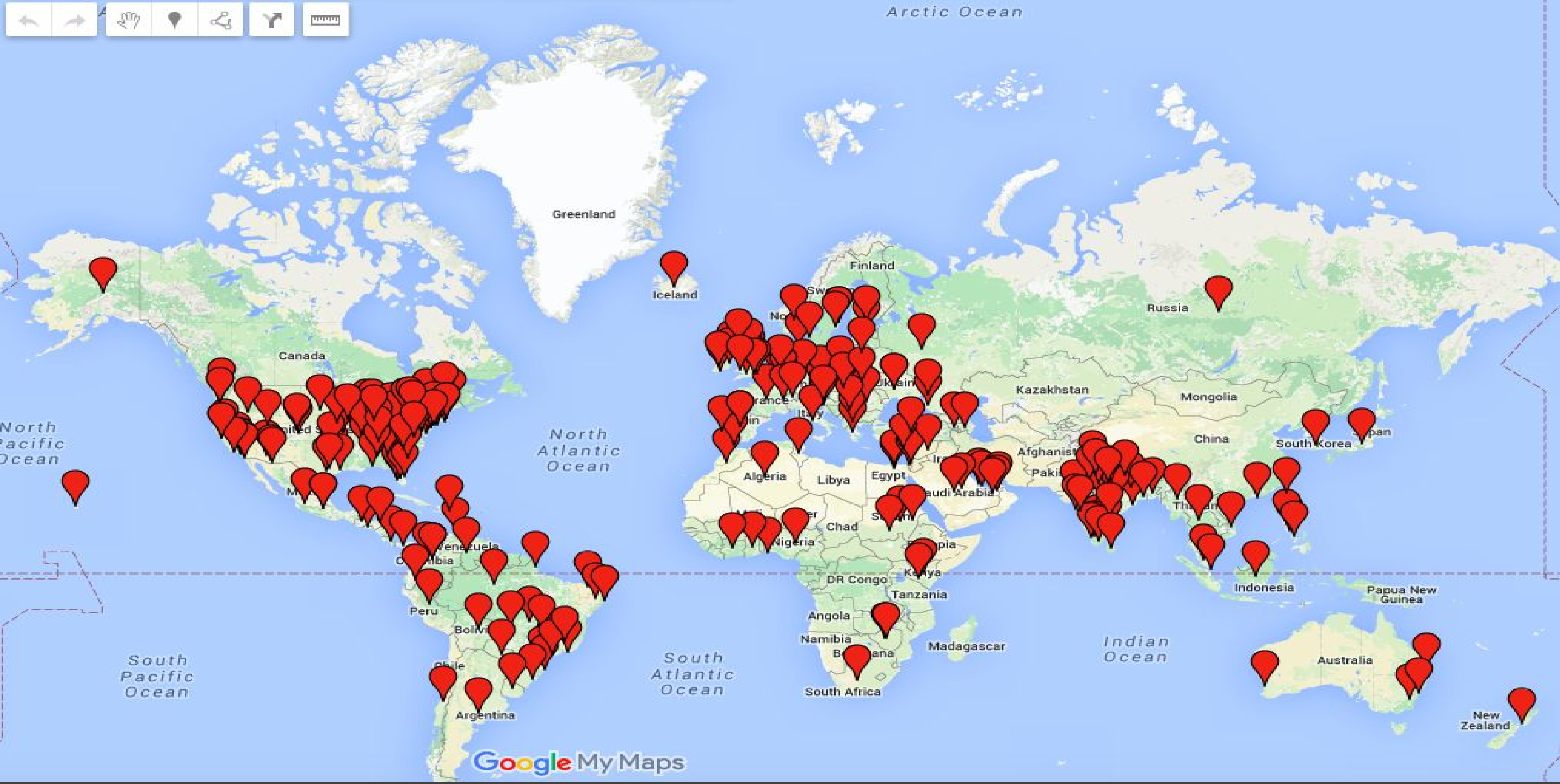
Employer supported volunteering can help a company’s:

- Reputation and credibility
- Recruitment and staff retention
- Staff morale and work performance
- Training and development
- Change management
- Government and regulatory relations..

*Reed Executive

16

Years of community service



276

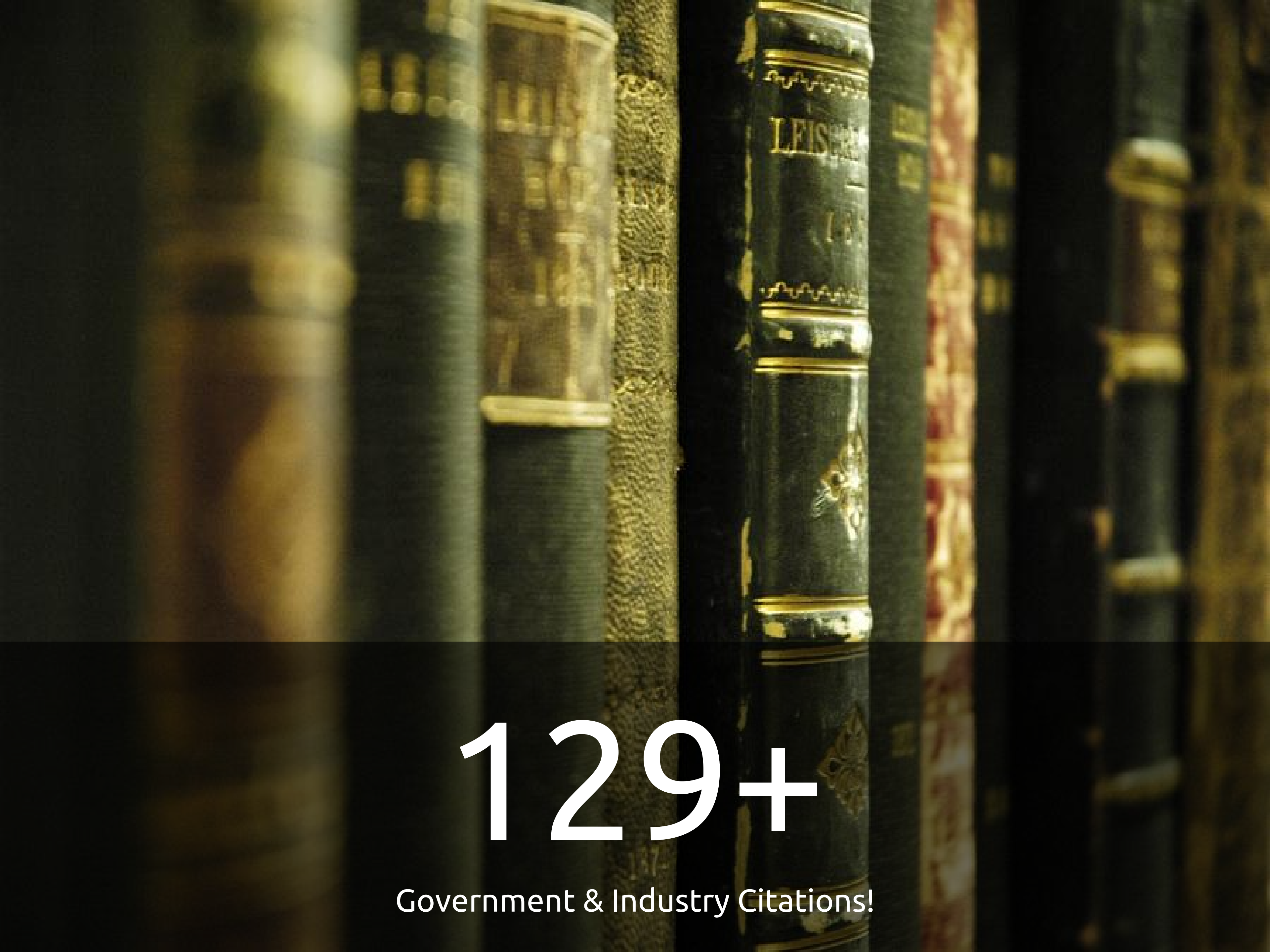
Active Chapters

93

Active Projects

55,000+

participants mailing lists



129+

Government & Industry Citations!

25

Academic Supporters

67

Paid Corporate Memberships

2406

Individual Members

2017 Board/Officers

- Chairperson: Matt Konda, Dallas, TX
- Vice Chairperson: Johanna Curiel, Curacao
- Secretary: Tom Brennan, New Jersey, USA
- Treasurer: Andrew van der Stock, Melbourne, Australia
- Board Member: Tobias Gondrom, Hong Kong
- Board Member: Michael Coates, San Francisco, CA, USA
- Board Member: Josh Sokol, Austin, TX

8 Employees

Employees

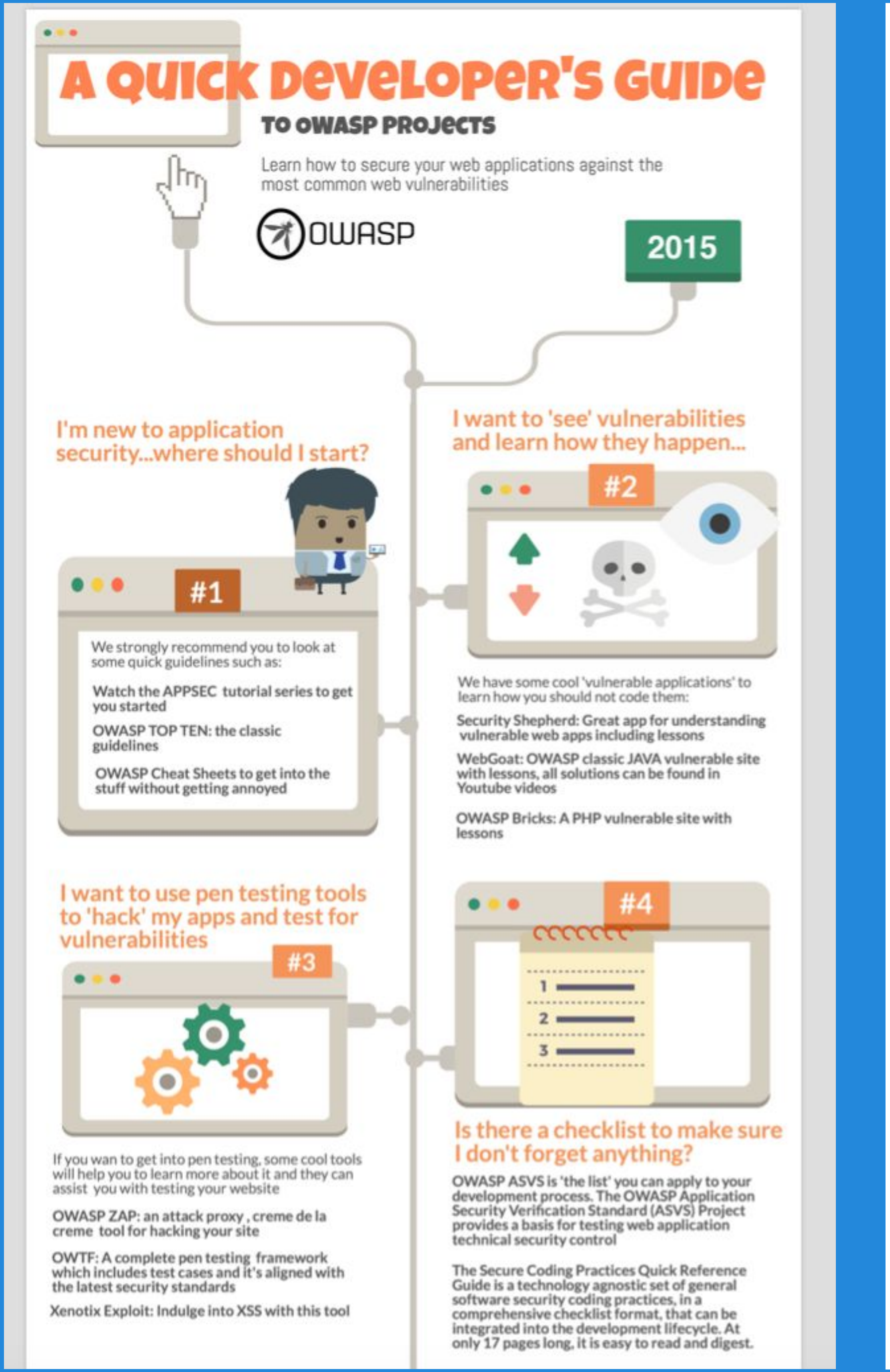
- Kate Hartmann, Operations Director,NJ - USA
- Matt Tesauro, Senior Project Coordinator, TX - USA
- Tiffany Long, Community Manager, CA - USA
- Kelly Santalucia, Membership and Business Liaison NJ - USA
- Claudia Casanovas, Project Coordinator, NJ - USA
- Alison Shrader, Accounting, MD – USA
- Laura Grau, Event Manager, CA – USA
- Dawn Aitken, Administrative Assistant, NJ - USA
- Hugo Costa, Graphic Design, (Contractor), Portugal
- ** help wanted!!

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

Contact Us

www.owasp.org

OWASP Projects



A QUICK DEVELOPER'S GUIDE

TO OWASP PROJECTS

Learn how to secure your web applications against the most common web vulnerabilities

I'm new to application security...where should I start?

We strongly recommend you to look at some quick guidelines such as:

Watch the APPSEC tutorial series to get you started

OWASP TOP TEN: the classic guidelines

OWASP Cheat Sheets to get into the stuff without getting annoyed

I want to 'see' vulnerabilities and learn how they happen...

We have some cool 'vulnerable applications' to learn how you should not code them:

Security Shepherd: Great app for understanding vulnerable web apps including lessons

WebGoat: OWASP classic JAVA vulnerable site with lessons, all solutions can be found in Youtube videos

OWASP Bricks: A PHP vulnerable site with lessons

I want to use pen testing tools to 'hack' my apps and test for vulnerabilities

If you wan to get into pen testing, some cool tools will help you to learn more about it and they can assist you with testing your website

OWASP ZAP: an attack proxy , creme de la creme tool for hacking your site

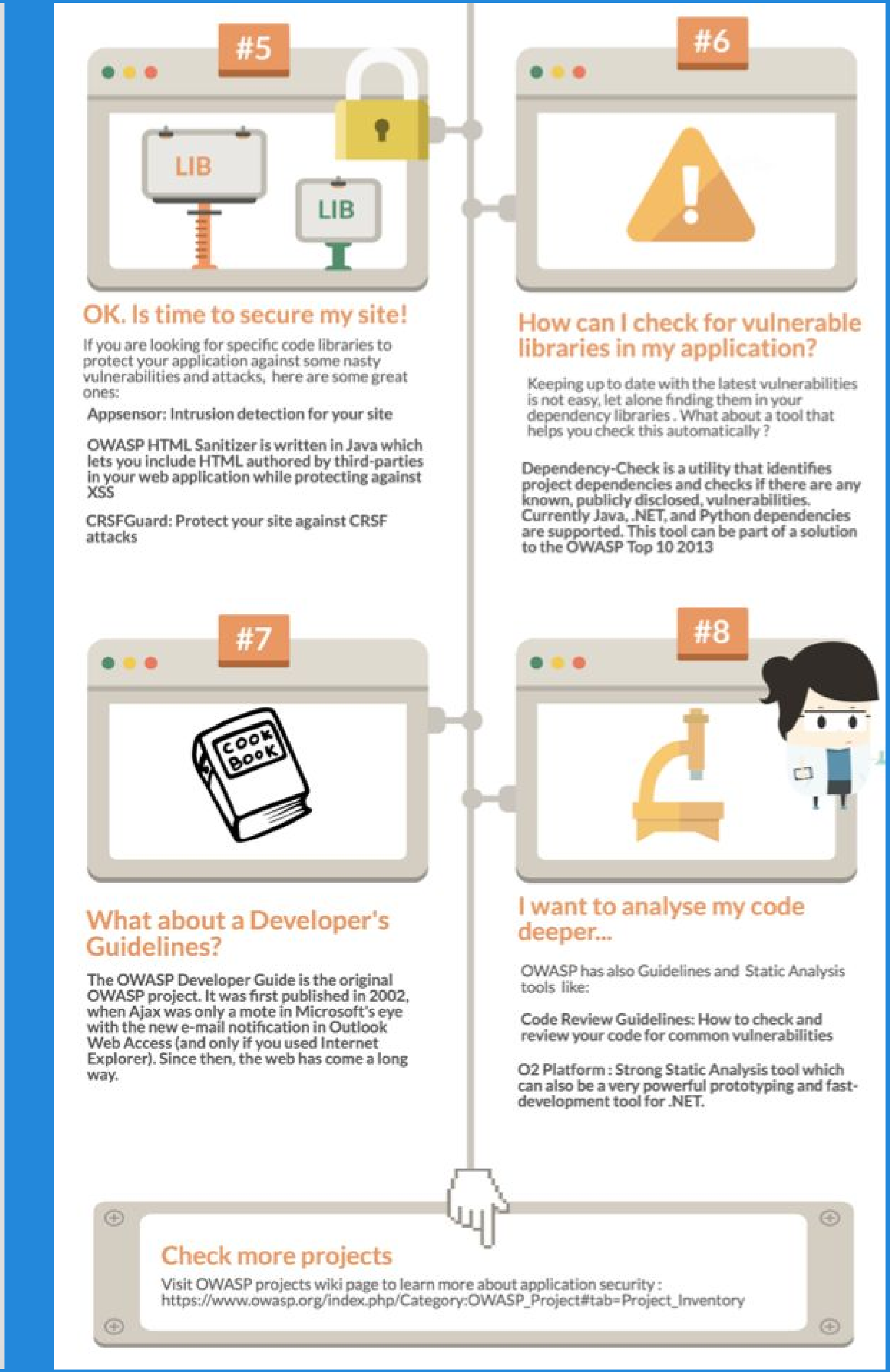
OWTF: A complete pen testing framework which includes test cases and it's aligned with the latest security standards

Xenotix Exploit: Indulge into XSS with this tool

Is there a checklist to make sure I don't forget anything?

OWASP ASVS is 'the list' you can apply to your development process. The OWASP Application Security Verification Standard (ASVS) Project provides a basis for testing web application technical security control

The Secure Coding Practices Quick Reference Guide is a technology agnostic set of general software security coding practices, in a comprehensive checklist format, that can be integrated into the development lifecycle. At only 17 pages long, it is easy to read and digest.



OK. Is time to secure my site!

If you are looking for specific code libraries to protect your application against some nasty vulnerabilities and attacks, here are some great ones:

Appsensor: Intrusion detection for your site

OWASP HTML Sanitizer is written in Java which lets you include HTML authored by third-parties in your web application while protecting against XSS

CRSFGuard: Protect your site against CRSF attacks

How can I check for vulnerable libraries in my application?

Keeping up to date with the latest vulnerabilities is not easy, let alone finding them in your dependency libraries . What about a tool that helps you check this automatically ?

Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities. Currently Java, .NET, and Python dependencies are supported. This tool can be part of a solution to the OWASP Top 10 2013

What about a Developer's Guidelines?

The OWASP Developer Guide is the original OWASP project. It was first published in 2002, when Ajax was only a mote in Microsoft's eye with the new e-mail notification in Outlook Web Access (and only if you used Internet Explorer). Since then, the web has come a long way.

I want to analyse my code deeper...

OWASP has also Guidelines and Static Analysis tools like:

Code Review Guidelines: How to check and review your code for common vulnerabilities

O2 Platform : Strong Static Analysis tool which can also be a very powerful prototyping and fast-development tool for .NET.

Check more projects

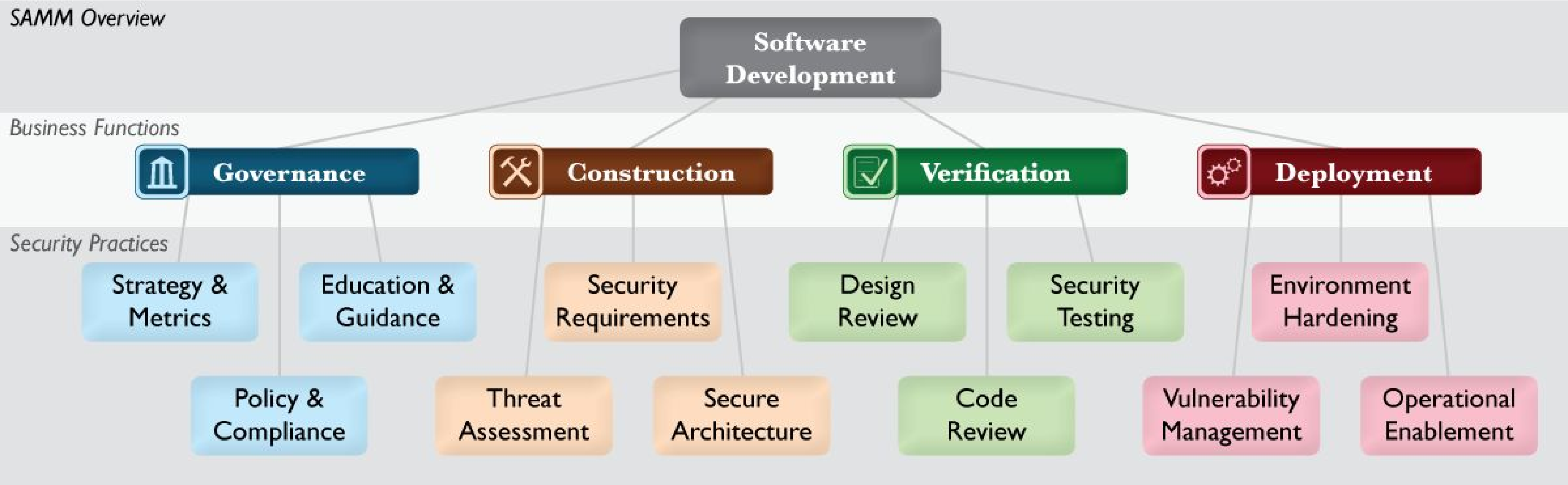
Visit OWASP projects wiki page to learn more about application security :
https://www.owasp.org/index.php/Category:OWASP_Project#tab=Project_Inventory

Flagship: Strategic value to OWASP, major review process to evaluate candidate projects

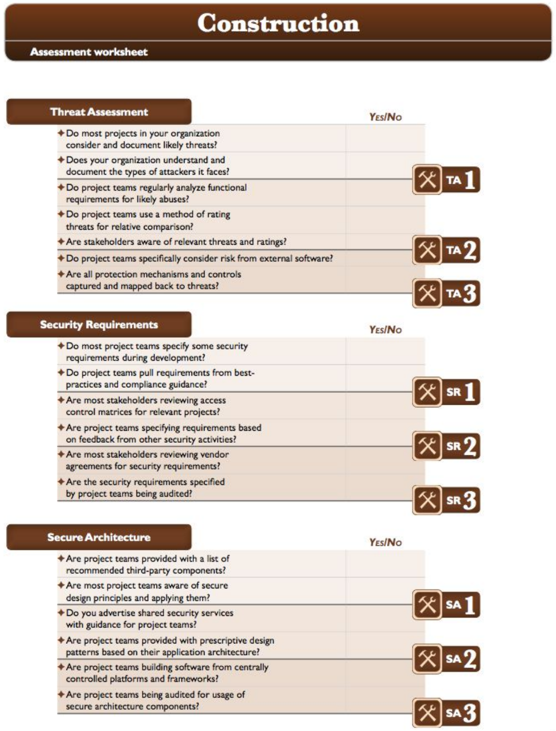
As of March 2017:

Tools	Code	Documentation
OWASP Zed Attack Proxy (ZAP)	OWASP ModSecurity Core Rule Set (CRS) Project	OWASP Application Security Verification Standard (ASVS) Project
OWASP Web Testing Environment Project	OWASP CSRFGuard Project	OWASP Software Assurance Maturity Model (SAMM)
OWASP OWTF	OWASP AppSensor Project	OWASP Top Ten Project
OWASP Dependency Check		OWASP Testing Project
OWASP Security Shepherd		OWASP AppSensor Project

https://www.owasp.org/index.php/Category:OWASP_Project#tab=Project_Inventory



OWASP SAMM



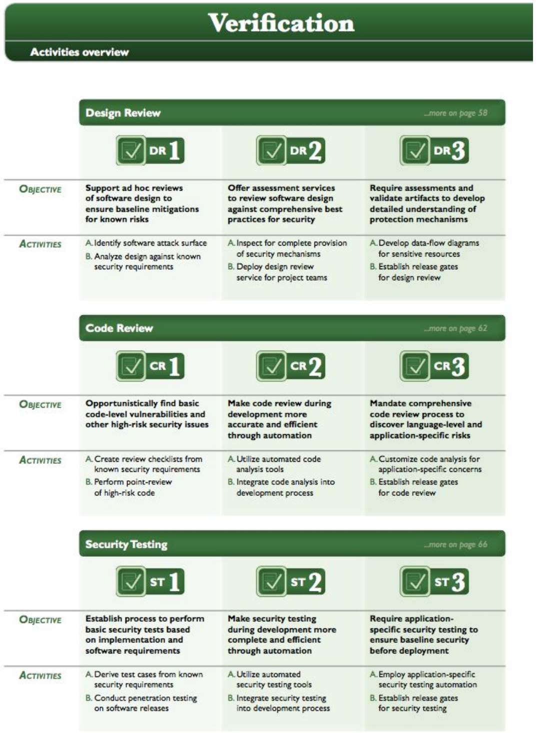
Construction

Assessment worksheet

Threat Assessment	Yes/No	
✦ Do most projects in your organization consider and document likely threats?		
✦ Does your organization understand and document the types of attackers it faces?		TA 1
✦ Do project teams regularly analyze functional requirements for likely abuses?		
✦ Do project teams use a method of rating threats for relative comparison?		
✦ Are stakeholders aware of relevant threats and ratings?		TA 2
✦ Do project teams specifically consider risk from external software?		
✦ Are all protection mechanisms and controls captured and mapped back to threats?		TA 3

Security Requirements	Yes/No	
✦ Do most project teams specify some security requirements during development?		
✦ Do project teams pull requirements from best-practices and compliance guidance?		SR 1
✦ Are most stakeholders reviewing access control matrices for relevant projects?		
✦ Are project teams specifying requirements based on feedback from other security activities?		SR 2
✦ Are most stakeholders reviewing vendor agreements for security requirements?		
✦ Are the security requirements specified by project teams being audited?		SR 3

Secure Architecture	Yes/No	
✦ Are project teams provided with a list of recommended third-party components?		
✦ Are most project teams aware of secure design principles and applying them?		SA 1
✦ Do you advertise shared security services with guidance for project teams?		
✦ Are project teams provided with prescriptive design patterns based on their application architecture?		SA 2
✦ Are project teams building software from centrally controlled platforms and frameworks?		
✦ Are project teams being audited for usage of secure architecture components?		SA 3



Verification

Activities overview

Design Review ...more on page 58

	DR 1	DR 2	DR 3
Objective	**Support ad hoc reviews of software design to ensure baseline mitigations for known risks**	**Offer assessment services to review software design against comprehensive best practices for security**	**Require assessments and validate artifacts to develop detailed understanding of protection mechanisms**
Activities	A. Identify software attack surface B. Analyze design against known security requirements	A. Inspect for complete provision of security mechanisms B. Deploy design review service for project teams	A. Develop data-flow diagrams for sensitive resources B. Establish release gates for design review

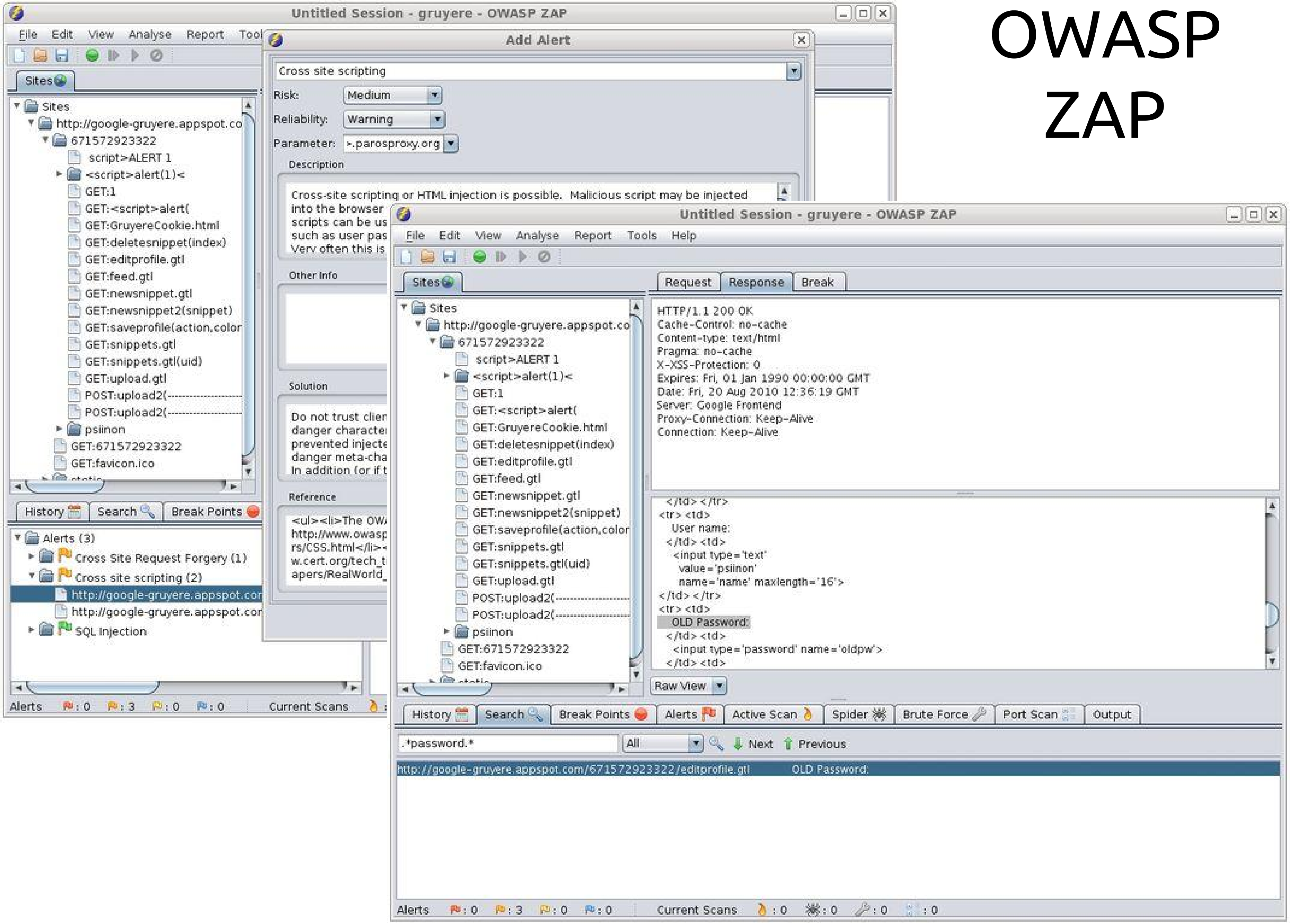
Code Review ...more on page 62

	CR 1	CR 2	CR 3
Objective	**Opportunistically find basic code-level vulnerabilities and other high-risk security issues**	**Make code review during development more accurate and efficient through automation**	**Mandate comprehensive code review process to discover language-level and application-specific risks**
Activities	A. Create review checklists from known security requirements B. Perform point-review of high-risk code	A. Utilize automated code analysis tools B. Integrate code analysis into development process	A. Customize code analysis for application-specific concerns B. Establish release gates for code review

Security Testing ...more on page 66

	ST 1	ST 2	ST 3
Objective	**Establish process to perform basic security tests based on implementation and software requirements**	**Make security testing during development more complete and efficient through automation**	**Require application-specific security testing to ensure baseline security before deployment**
Activities	A. Derive test cases from known security requirements B. Conduct penetration testing on software releases	A. Utilize automated security testing tools B. Integrate security testing into development process	A. Employ application-specific security testing automation B. Establish release gates for security testing

OWASP ZAP



Cross site scripting

Risk: Medium

Reliability: Warning

Parameter: >.parosproxy.org

Description

Cross-site scripting or HTML injection is possible. Malicious script may be injected into the browser
scripts can be us
such as user pas
Very often this is

Other Info

Solution

Do not trust clien
danger characte
prevented injecte
danger meta-cha
In addition (or if t

Reference

<ul><li>The OW
http://www.owasp
rs/CSS.html</li><
w.cert.org/tech_ti
apers/RealWorld_

Sites
- http://google-gruyere.appspot.co
  - 671572923322
    - script>ALERT 1
    - <script>alert(1)<
    - GET:1
    - GET:<script>alert(
    - GET:GruyereCookie.html
    - GET:deletesnippet(index)
    - GET:editprofile.gtl
    - GET:feed.gtl
    - GET:newsnippet.gtl
    - GET:newsnippet2(snippet)
    - GET:saveprofile(action,color
    - GET:snippets.gtl
    - GET:snippets.gtl(uid)
    - GET:upload.gtl
    - POST:upload2(-----------------
    - POST:upload2(-----------------
    - psiinon
  - GET:671572923322
  - GET:favicon.ico
  - static

Alerts (3)
- Cross Site Request Forgery (1)
- Cross site scripting (2)
  - http://google-gruyere.appspot.com
  - http://google-gruyere.appspot.com
- SQL Injection

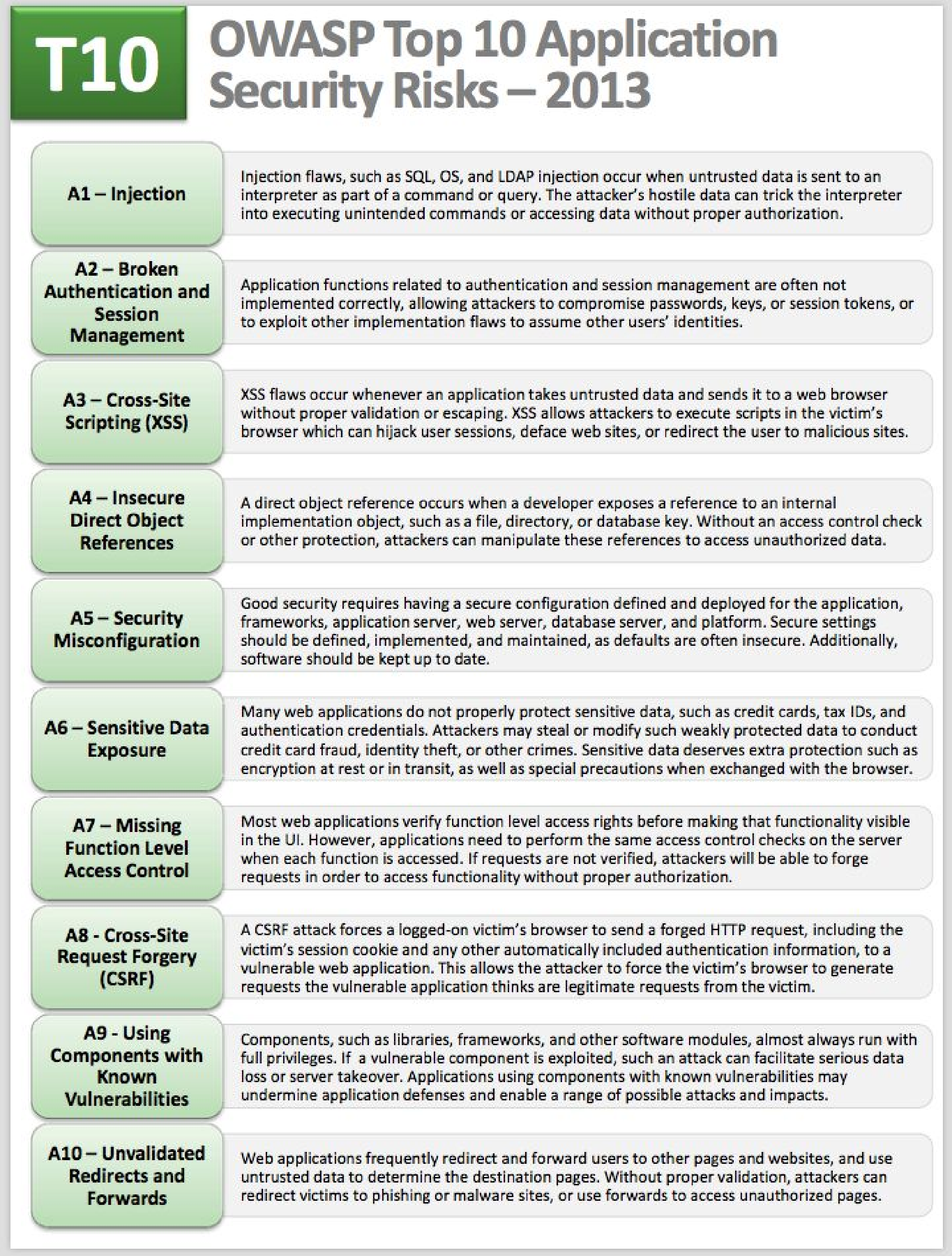
```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-type: text/html
Pragma: no-cache
X-XSS-Protection: 0
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Date: Fri, 20 Aug 2010 12:36:19 GMT
Server: Google Frontend
Proxy-Connection: Keep-Alive
Connection: Keep-Alive
```

```
  </td></tr>
<tr><td>
   User name:
  </td><td>
    <input type='text'
     value='psiinon'
     name='name' maxlength='16'>
</td></tr>
<tr><td>
   OLD Password:
  </td><td>
    <input type='password' name='oldpw'>
  </td><td>
```

.*password.* All

http://google-gruyere.appspot.com/671572923322/editprofile.gtl OLD Password:

Alerts 🏴: 0 🏴: 3 🏴: 0 🏴: 0 Current Scans : 0 : 0 : 0 : 0



T10 OWASP Top 10 Application Security Risks – 2013

A1 – Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2 – Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

A3 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A4 – Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

A5 – Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

A6 – Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

A7 – Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

A8 - Cross-Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

A9 - Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

A10 – Unvalidated Redirects and Forwards

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Lab: Projects that have produced a deliverable of value. Expectation of producing releases that are ready for mainstream usage.

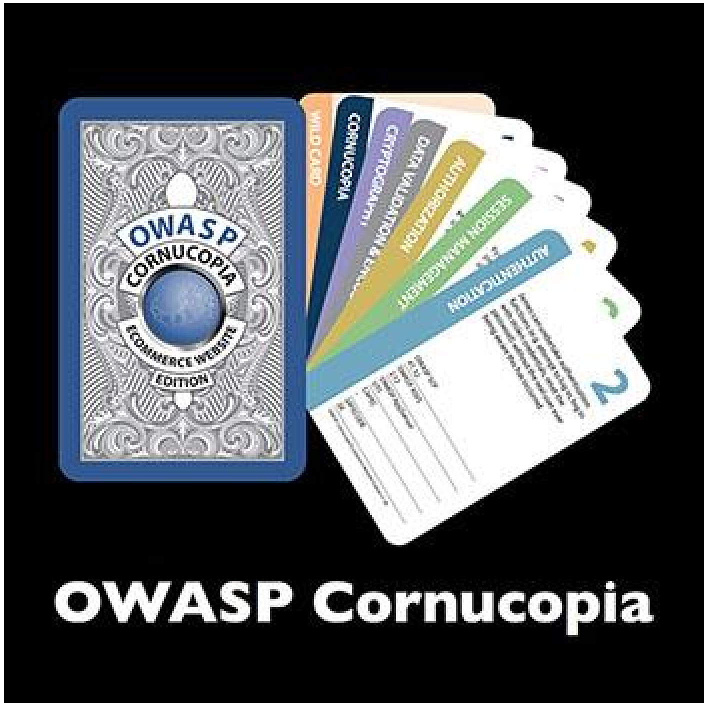
Some examples (as of March 2017):

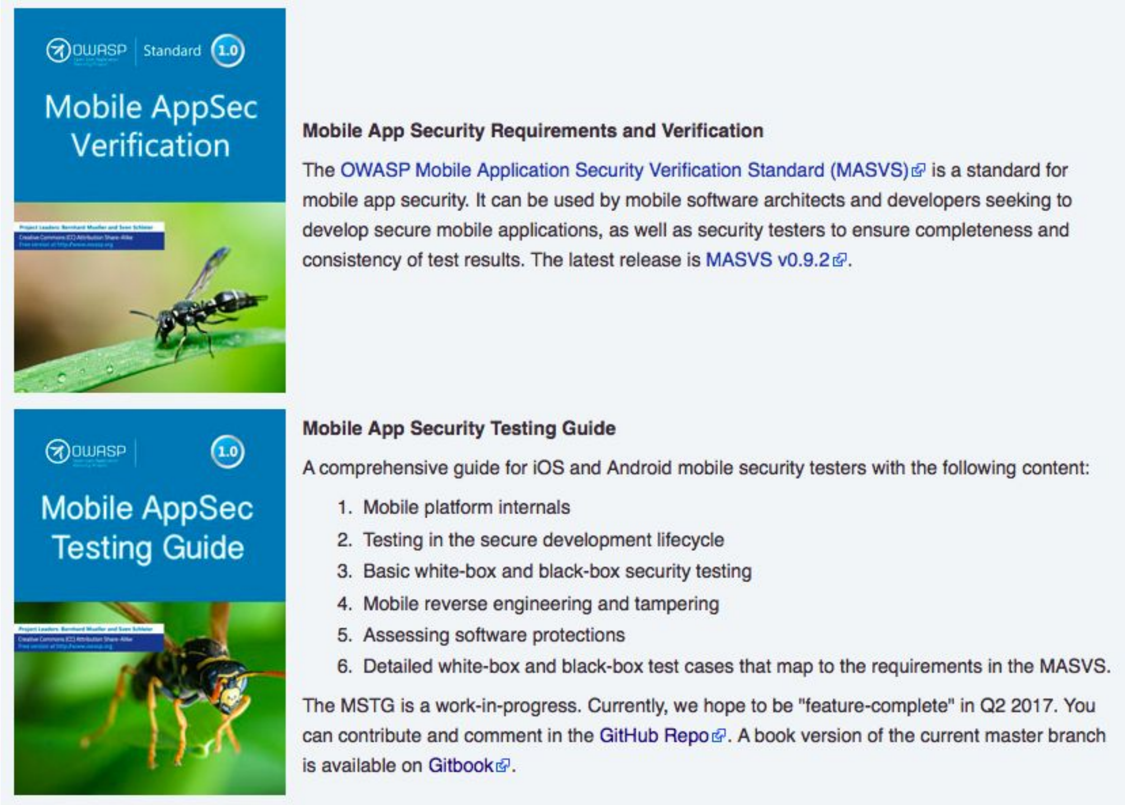
OWASP WebGoat Project

OWASP Cornucopia

OWASP Mobile Security Project

OWASP Enterprise Security API (ESAPI)





Mobile App Security Requirements and Verification

The OWASP Mobile Application Security Verification Standard (MASVS) is a standard for mobile app security. It can be used by mobile software architects and developers seeking to develop secure mobile applications, as well as security testers to ensure completeness and consistency of test results. The latest release is MASVS v0.9.2.

Mobile App Security Testing Guide

A comprehensive guide for iOS and Android mobile security testers with the following content:

1. Mobile platform internals
2. Testing in the secure development lifecycle
3. Basic white-box and black-box security testing
4. Mobile reverse engineering and tampering
5. Assessing software protections
6. Detailed white-box and black-box test cases that map to the requirements in the MASVS.

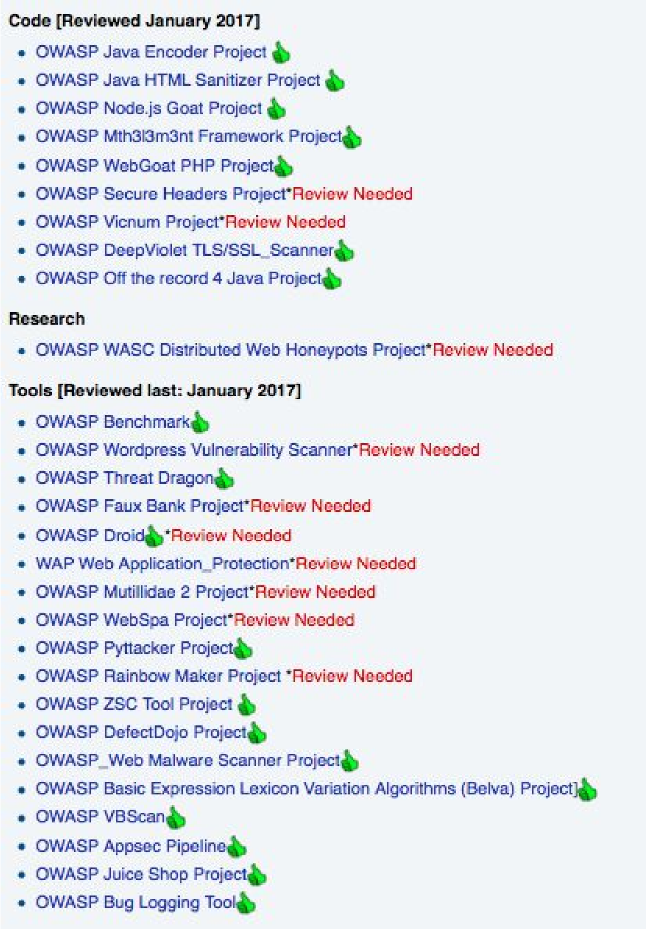
The MSTG is a work-in-progress. Currently, we hope to be "feature-complete" in Q2 2017. You can contribute and comment in the GitHub Repo. A book version of the current master branch is available on Gitbook.

OWASP

INCUBATOR

new projects

Incubator: Projects that are still maturing.
As of March 2017:



Code [Reviewed January 2017]

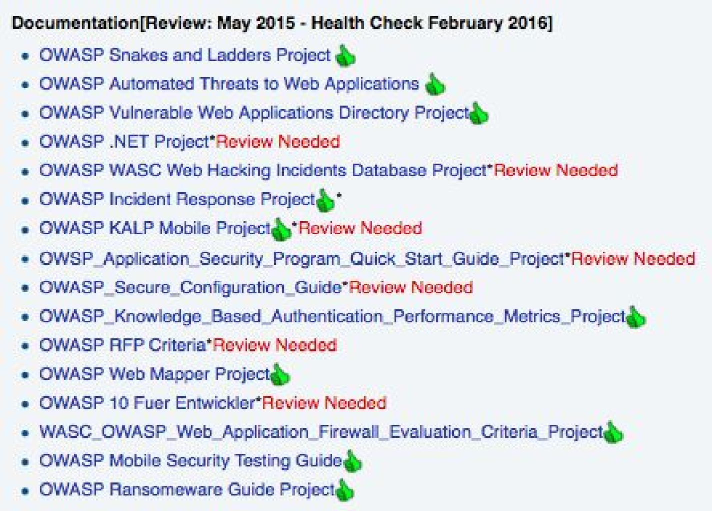
- OWASP Java Encoder Project 👍
- OWASP Java HTML Sanitizer Project 👍
- OWASP Node.js Goat Project 👍
- OWASP Mth3l3m3nt Framework Project👍
- OWASP WebGoat PHP Project👍
- OWASP Secure Headers Project*Review Needed
- OWASP Vicnum Project*Review Needed
- OWASP DeepViolet TLS/SSL_Scanner👍
- OWASP Off the record 4 Java Project👍

Research

- OWASP WASC Distributed Web Honeypots Project*Review Needed

Tools [Reviewed last: January 2017]

- OWASP Benchmark👍
- OWASP Wordpress Vulnerability Scanner*Review Needed
- OWASP Threat Dragon👍
- OWASP Faux Bank Project*Review Needed
- OWASP Droid👍*Review Needed
- WAP Web Application_Protection*Review Needed
- OWASP Mutillidae 2 Project*Review Needed
- OWASP WebSpa Project*Review Needed
- OWASP Pyttacker Project👍
- OWASP Rainbow Maker Project *Review Needed
- OWASP ZSC Tool Project 👍
- OWASP DefectDojo Project👍
- OWASP_Web Malware Scanner Project👍
- OWASP Basic Expression Lexicon Variation Algorithms (Belva) Project]👍
- OWASP VBScan👍
- OWASP Appsec Pipeline👍
- OWASP Juice Shop Project👍
- OWASP Bug Logging Tool👍



Documentation[Review: May 2015 - Health Check February 2016]

- OWASP Snakes and Ladders Project 👍
- OWASP Automated Threats to Web Applications 👍
- OWASP Vulnerable Web Applications Directory Project👍
- OWASP .NET Project*Review Needed
- OWASP WASC Web Hacking Incidents Database Project*Review Needed
- OWASP Incident Response Project👍*
- OWASP KALP Mobile Project👍*Review Needed
- OWSP_Application_Security_Program_Quick_Start_Guide_Project*Review Needed
- OWASP_Secure_Configuration_Guide*Review Needed
- OWASP_Knowledge_Based_Authentication_Performance_Metrics_Project👍
- OWASP RFP Criteria*Review Needed
- OWASP Web Mapper Project👍
- OWASP 10 Fuer Entwickler*Review Needed
- WASC_OWASP_Web_Application_Firewall_Evaluation_Criteria_Project👍
- OWASP Mobile Security Testing Guide👍
- OWASP Ransomeware Guide Project👍

Participating in a project ...

OWASP Projects are **community driven** and most projects are **open** for anyone motivated to join!

Get involved!

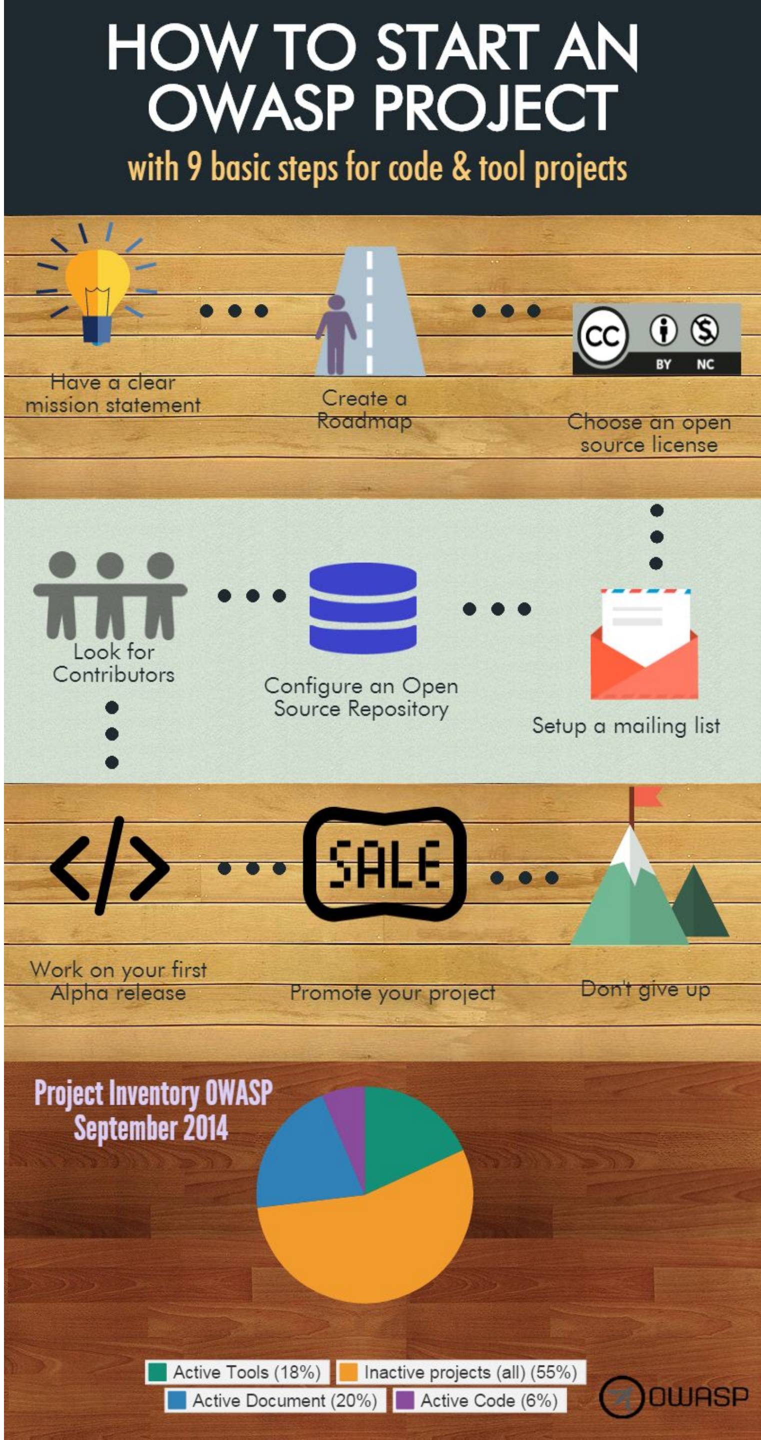
Join the mailing list, get in touch, contact the project leader, test the software / documentation, report bugs, propose features, etc.

Starting a project …

Review the OWASP Project Inventory for existing projects

Follow the 2016 OWASP Project Process - Workflow …

https://www.owasp.org/index.php/Category:OWASP_Project#tab=Starting_a_New_Project


HOW TO START AN OWASP PROJECT
with 9 basic steps for code & tool projects
Have a clear mission statement
Create a Roadmap
Choose an open source license
BY NC
Look for Contributors
Configure an Open Source Repository
Setup a mailing list
Work on your first Alpha release
SALE
Promote your project
Don't give up
Project Inventory OWASP September 2014
Active Tools (18%)
Inactive projects (all) (55%)
Active Document (20%)
Active Code (6%)
OWASP

Thank you!

"merci" French Country: Can…a (Quebec)
…kka þ… …elandi… …ountry Iceland
"…heers" E…glish C…ntry UK
Nor…egia… Co…ntry: Norway
"dz…kuję" P…h Cou…y Pola…
ευχαριστώ "eycharisto" Gre… Countr… Greec…
сп…ибо "sp…ibo" R…ian Country: Russia
谢谢 "xièxie" Mandarin Country: China
ありがとう "arigatou" Japanese Country: Japan
شكرا "shukran" Arabic Country: Egypt
"thanks" English Country: US
"mulțumesc" Romanian Country: Romania
شکریہ "shukria" Punjabi Country: Pakistan
"di ou mèsi" Haitian Creole Country: Haiti
"O se" Yoruba Country: Nigeria
ممنون "mamnoon" Persian Country: Iran
ขอบคุณ "khopkhun" Thai Country: Thailand
"dank u" Dutch Country: Suriname
"dankie" Afrikaans Country: South Africa
धन्यवाद "dhanyavād" Hindi Country: India
"gracias" Spanish Country: Argentina
"terima kasih" Indonesian Country: Indonesia
"ta" English Country: Australia
"obrigado" Portuguese Country: Brazil
"asante" Swahili Country: Kenya
"teşekkür ederim" Turkish Country: Turkey