Introduction to the OWASP Top 10

Austin Chamberlain, OWASP Auckland

Top 10 Background

- Started in 2003
- Awareness document, targeted at:
  - Developers
  - Security professionals
  - Security governance and management
- **Lists the Most Critical Security Risks to Web Applications**
- Risks - not exploits or impacts
- OWASP Flagship Project

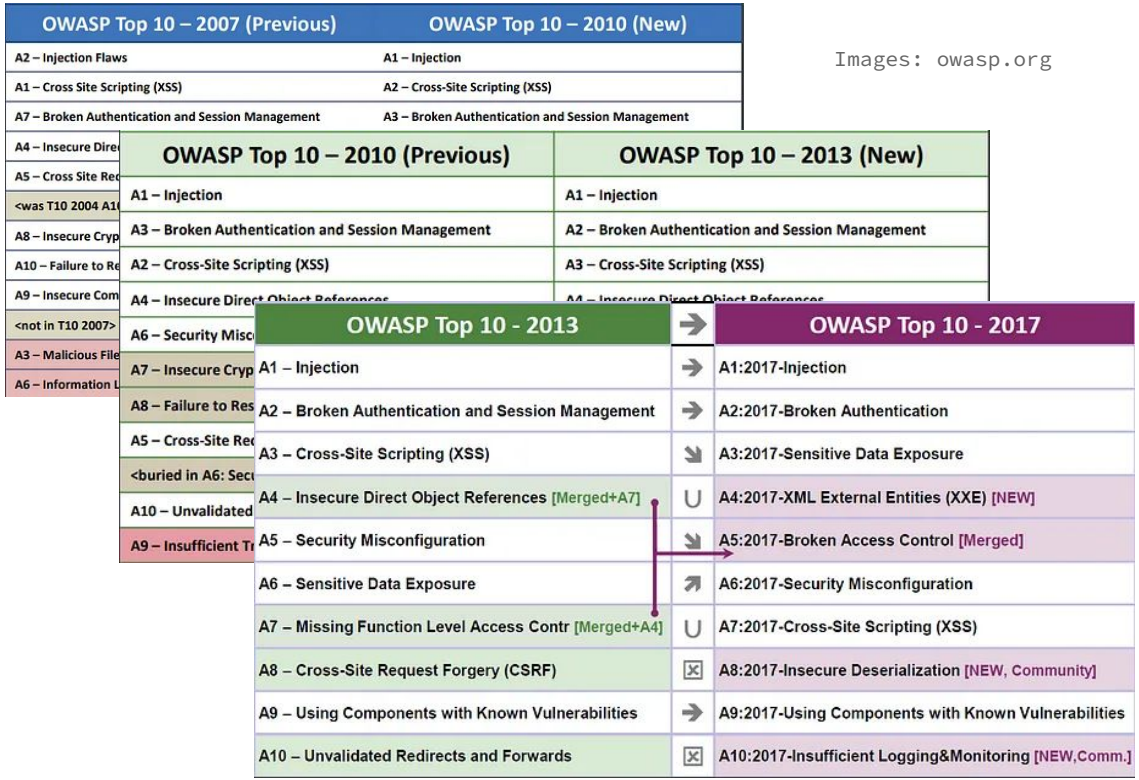
Versions and Updates

2004, 2007, 2010, 2013

2017

2021 – current version

2025 – under development



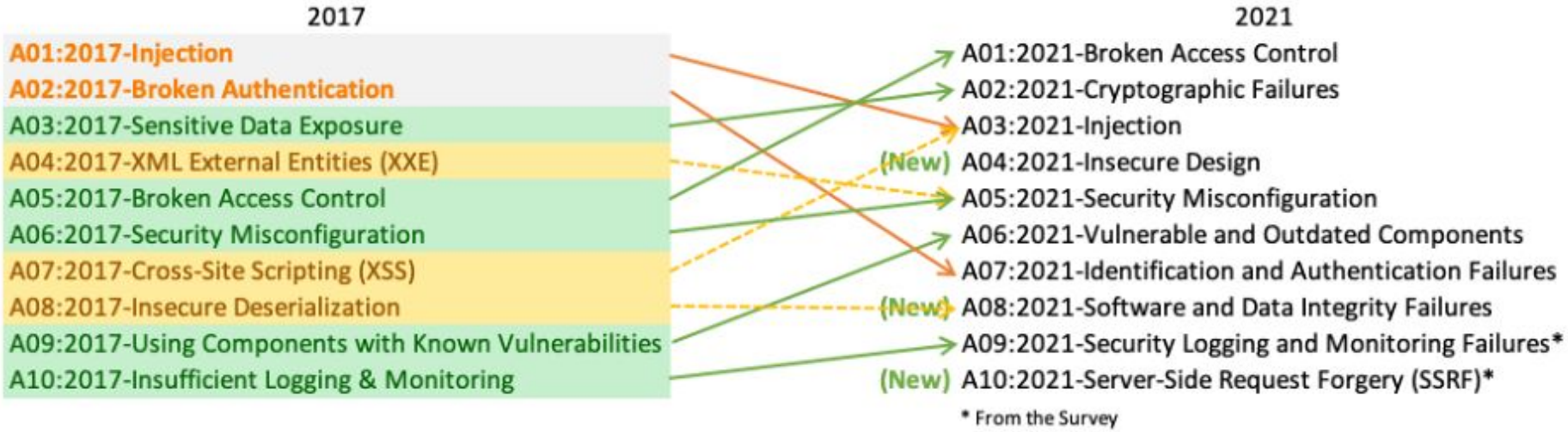
OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection Flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross-Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Images: owasp.org

2021 Top 10



Images: owasp.org

3 new categories, 4 categories with name/scope changes, some consolidation

2025 Top 10

OWASP Top Ten 2025

Current project status as of July, 2024
We are planning to announce the release of the OWASP Top 10:2025 in early 2025.
https://owasp.org/Top10

Data Collection (Now - Dec 2024)

Industry Survey (Drafting)

Data Normalization (Pending)

Review Process (TBD)

Documentation Updates (TBD)

International Translations

Methodology

Hybrid system - data and survey

- Eight categories from contributed data
- Two categories from community survey

Why? Completeness.

Data results generally limited to automated tests, which take time to develop and refine.

Community survey allows front line experts to highlight issues not yet in the data.

Image: DALL-E 3

A01: Broken Access Control

Access control failure allows a function outside user's intended limits.

- Bypass access control
- Access another account
- Elevation of privilege
- Violation of least privilege/default deny

Prevention:

Deny by default; unified access control across application; limit metadata and rates

Image: "Desire Path", flickr.com

A02: Cryptographic Failures

Causes:

- Cleartext protocols - HTTP, FTP
- Old, weak, or deprecated algorithms - MD5
- Default keys, weak keys, key management (are your keys in your Git repo?)
- Server certificate correct and validated?

Prevention:

- Store only required sensitive data, and classify appropriately
- Use up-to-date algorithms and protocols, and manage keys
- Encrypt data at rest and in transit
- Ensure cryptographic randomness is applied where required
- DO NOT ROLL YOUR OWN CRYPTO

A03: Injection

Drops to third position, even with inclusion of XSS

Causes:

- User-supplied data is not validated or sanitized
- Hostile data is directly used

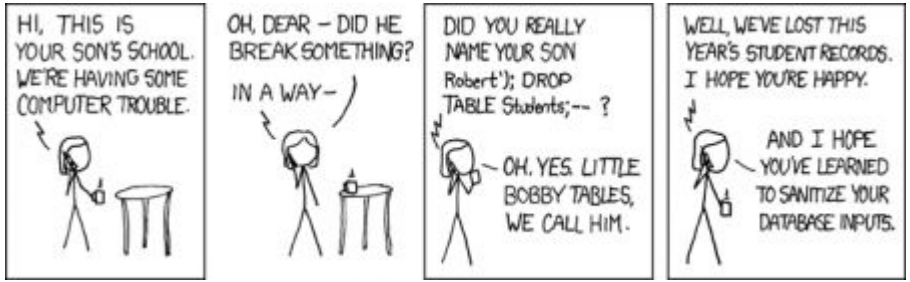
Examples:

- SQL, OS Command

Prevention:

- Safe API
- That's it.
- OK, server-side input validation - but get it right!
- LIMIT in SQL queries

Image: xkcd.com



A04: Insecure Design

New for 2021 - design and architectural flaws.

“Shift left” beyond coding to pre-code design.

Examples:

- Bots for ticket/item scalping
- Booking/ordering system attacks

Prevention:

- Secure development life cycle
- Threat modeling
- OWASP SAMM (Security Assurance Maturity Model)

Image: @karinakovacs2

A05: Security Misconfiguration

Causes:

- Cloud services permissions
- Unnecessary features
- Default accounts
- Verbose error messages
- Security settings not correctly applied

Prevention:

- Ongoing security testing and hardening
- Minimal platforms
- Segmented architecture - eg containerization, cloud security groups

Image: DALL-E 3

A06: Vulnerable and Outdated Components

Qualifies for OWASP top 10 from both community survey and data!

Does not directly link to CVEs (Common Vulnerability and Exposure)

Examples:

- Unaware of supply chain or software bill of materials (SBOM)
- Software components vulnerable or unsupported
- No or slow upgrades, scanning

Prevention:

- Removed unused components
- Have a software inventory and continuously update!
- Manage software updates over correct channels
- If old software must be used, mitigate and document.

A07: Identification and Authentication Failures

Previously *Broken Authentication*

Causes - does application allow:

- Credential stuffing
- Brute force attacks
- Weak passwords

Does application:

- Lack Multi-Factor Authentication (MFA)?
- Not hash stored passwords?
- Expose or reuse session ID?

Prevention:

- MFA
- Rate-limit and monitor for systematic password attacks
- Enforce strong passwords
- … and don’t allow default creds!
- Prevent account enumeration through standard messages

A08: Software and Data Integrity Failures

Examples:

- Application relies on plugins, libraries and modules from untrusted sources, repositories, and content delivery networks (CDN)
- Insecure CI/CD pipeline
- Attackers are using typo-squatting attacks against common module names

Prevention:

- Use signed software channels.
- Use trusted repositories, or even internal repositories.
- Use a software supply chain security tool - OWASP Dependency Check or OWASP CycloneDX
- Implement review process for code and configuration changes.

A09: Security Logging and Monitoring Failures

Would you detect an attack? With enough time to react? With enough information to respond?

- Are you logging and monitoring?
- What kind of events?
- Is someone actually reading the logs?
- Where are the logs stored?
- Are there alerts on the logs?
- Would a pentest trigger an alert?

Prevention:

- Log access events with context and good retention period.
- Generate logs in useful format.
- Protect logs against injections/attacks.
- Institute effective monitoring and alerting.
- Have an incident response plan.

A10: Server Side Request Forgery

Category from community survey.

Low incidence rate in data, above-average Exploit/Impact potential rating.

SSRF definition: when a web application fetches a remote resource without validating the user-supplied URL.

Architecture complexity and cloud services increase SSRF severity.

Prevention:

Network: segmentation, firewall “deny by default”

Application: sanitize client-supplied data, do not send raw responses to clients

OWASP Top Ten - Quick Lessons

1. OWASP Top 10 is not everything! There are other risks.
2. Frameworks solve a lot of problems.
3. Threat modeling (especially at the start) and testing (always).
4. Other OWASP projects provide specific guidance on development, verification, and testing.