

# Basic Principles and Resources

Saltzer & Schröder, IEEE CSD,  
OWASP Top 10 / STG / ASVS

# Rough Overview

1. Introduction
2. >> Basic Principles and Resources <<
3. Architecture & Basic Web Procedure
4. Authentication and Session Management
5. Authorization
6. Server and Backend Attacks
7. Remaining Client Attacks
8. General Topics
9. Conclusions

There are a few basic principles that  
apply to almost all vulnerabilities

Economy of Mechanism

---

Fail-safe Defaults

---

Complete Mediation

---

Least Privilege

---

Least Common Mechanism

---

Separation of Privilege

---

Open Design

---

Psychological Acceptability

- Saltzer and Schroeder, 1975 -

[http://web.cs.wpi.edu/~guttman/cs557\\_website/papers/saltzer1975.pdf](http://web.cs.wpi.edu/~guttman/cs557_website/papers/saltzer1975.pdf)

<https://adam.shostack.org/blog/the-security-principles-of-saltzer-and-schroeder/>

Earn or give, but never assume, trust.

---

Use an authentication mechanism that cannot be bypassed or tampered with.

---

Authorize after you authenticate

---

Strictly separate data and control instructions, and never process control instructions received from untrusted sources.

---

Define an approach that ensures all data are explicitly validated.

---

Use cryptography correctly.

---

Identify sensitive data and how they should be handled.

---

Always consider the user.

---

Understand how integrating external components changes your attack surface.

---

Be flexible when considering future changes to objects and actors.

- IEEE Center for Secure Design, 2014 -

<https://ieeecs-media.computer.org/media/technical-activities/CYBSI/docs/Top-10-Flaws.pdf>

Some resources which might be  
useful...

# OWASP

(Open Web Application Security Project)

# OWASP Top 10

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.]

<https://owasp.org/www-project-top-ten/>



# OWASP ASVS

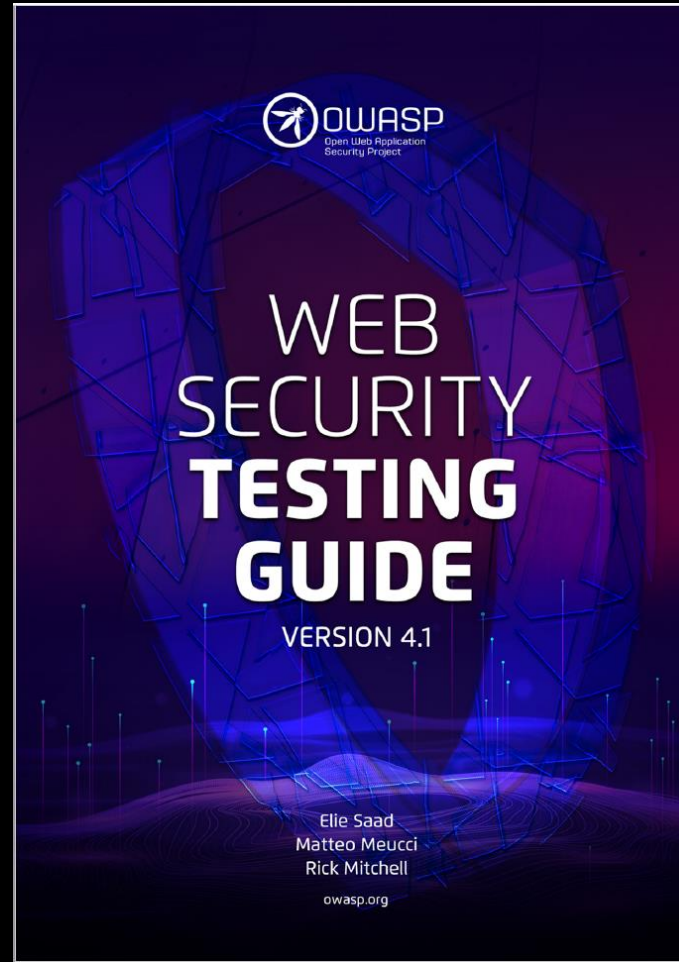
- V1: Architecture, Design and Threat Modeling Requirements
- V2: Authentication Verification Requirements
- V3: Session Management Verification Requirements
- V4: Access Control Verification Requirements
- V5: Validation, Sanitization and Encoding Verification Requirements
- V6: Stored Cryptography Verification Requirements
- V7: Error Handling and Logging Verification Requirements
- V8: Data Protection Verification Requirements
- V9: Communications Verification Requirements
- V10: Malicious Code Verification Requirements
- V11: Business Logic Verification Requirements
- V12: File and Resources Verification Requirements
- V13: API and Web Service Verification Requirements
- V14: Configuration Verification Requirements

## V3.2 Session Binding Requirements

#	Description	L1	L2	L3	CWE	<a href="#">NIST</a> <a href="#">§</a>
3.2.1	Verify the application generates a new session token on user authentication. <a href="#">(C6)</a>	✓	✓	✓	384	7.1
3.2.2	Verify that session tokens possess at least 64 bits of entropy. <a href="#">(C6)</a>	✓	✓	✓	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. <a href="#">(C6)</a>		✓	✓	331	7.1

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

# OWASP Web Security Testing Guide



<https://owasp.org/www-project-web-security-testing-guide/>

# Some more interesting resources

- Similar courses from other universities
  - Feross Aboukhadijeh (Stanford)
    - <https://web.stanford.edu/class/cs253/>
  - Andreas Happe (FH/Technikum Wien)
    - <https://snikt.net/WebSec.pdf>
  - Björn Kimminich
    - <https://github.com/bkimminich/it-security-lecture>
- PortSwigger Web Security Academy
  - <https://portswigger.net/web-security>
- Hacker101
  - <https://www.hacker101.com/>

# Key Messages

- Sticking to a few design principles helps to avoid a lot of flaws and vulnerabilities
  - they also help to find flaws and vulnerabilities
- A lot of great resources are available when it comes to web security
- OWASP is your friend