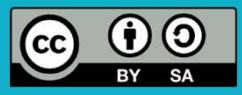
FH JOANNEUM
Elektronik und
Computer Engineering

Security and Testing
Knowledge for Security Testing

FH JOANNEUM
Institute of Electronic Engineering
System Test Engineering
Egon Teiniker
Version: 1.5.0



Knowledge for Security Testing

Outline

- Introduction
- OWASP IoT Top 10
- OWASP API Security Top 10 - 2023
- OWASP Top 10
- CWE/SANS Top 25

Knowledge for Security Testing

Introduction

Critical software security knowledge and expertise can be compiled:

- **OWASP Top 10 (for IoT and web applications)**
- **CWE/SANS Top 25**

These lists are a **tool for education and awareness** to help programmers to prevent the kinds of vulnerabilities by identifying and avoiding all-too-common mistakes that occur **before software is even shipped**.

Knowledge for Security Testing

OWASP IoT Top 10 – 2018

- **I01:2018 - Weak, Guessable, or Hardcoded Passwords**
  Use of easily brute forced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.

- **I02:2018 - Insecure Network Services**
  Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control.

Knowledge for Security Testing

OWASP IoT Top 10 – 2018

- **I03:2018 - Insecure Ecosystem Interfaces**
  Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.

- **I04:2018 - Lack of Secure Update Mechanism** Tricky for regular IoT devices.
  Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.

Knowledge for Security Testing

OWASP IoT Top 10 – 2018

- **I05:2018 - Use of Insecure or Outdated Components**
  Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain.

- **I06:2018 - Insufficient Privacy Protection**
  User's personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.

Knowledge for Security Testing

OWASP IoT Top 10 – 2018

- **I07:2018 - Insecure Data Transfer and Storage**
  Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.

- **I08:2018 - Lack of Device Management**
  Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.

Knowledge for Security Testing

OWASP IoT Top 10 – 2018

- **I09:2018 - Insecure Default Settings**
  Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

- **I10:2018 Lack of Physical Hardening**
  Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device. Hardware Hacking, we can access the UART interface....

Knowledge for Security Testing

OWASP API Security Top 10 - 2023

- **API1:2023 - Broken Object Level Authorization**
  APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface of Object Level Access Control issues. Object level authorization checks should be considered in every function that accesses a data source using an ID from the user.

- **API2:2023 - Broken Authentication**
  Authentication mechanisms are often implemented incorrectly, allowing attackers to compromise authentication tokens or to exploit implementation flaws to assume other user's identities temporarily or permanently.

  For example getting anallowed accessed to python functionality.

Knowledge for Security Testing

OWASP API Security Top 10 - 2023

- **API3:2023 - Broken Object Property Level Authorization**
  This category combines API3:2019 Excessive Data Exposure and API6:2019 - Mass Assignment, focusing on the root cause: the lack of or improper authorization validation at the object property level. This leads to information exposure or manipulation by unauthorized parties.

- **API4:2023 - Unrestricted Resource Consumption** For example sending to much data...
  Satisfying API requests requires resources such as network bandwidth, CPU, memory, and storage. Other resources such as emails/SMS/phone calls or biometrics validation are made available by service providers via API integrations, and paid for per request. Successful attacks can lead to Denial of Service or an increase of operational costs.

Knowledge for Security Testing

OWASP API Security Top 10 - 2023

- **API5:2023 - Broken Function Level Authorization**
  Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, tend to lead to authorization flaws. By exploiting these issues, attackers can gain access to other users' resources and/or administrative functions.

- **API6:2023 - Unrestricted Access to Sensitive Business Flows**
  APIs vulnerable to this risk expose a business flow - such as buying a ticket, or posting a comment - without compensating for how the functionality could harm the business if used excessively in an automated manner. This doesn't necessarily come from implementation bugs.

Knowledge for Security Testing

OWASP API Security Top 10 - 2023

- **API7:2023 - Server Side Request Forgery**
  Server-Side Request Forgery (SSRF) flaws can occur when an API is fetching a remote resource without validating the user-supplied URI. This enables an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall or a VPN.

- **API8:2023 - Security Misconfiguration**
  APIs and the systems supporting them typically contain complex configurations, meant to make the APIs more customizable. Software and DevOps engineers can miss these configurations, or don't follow security best practices when it comes to configuration, opening the door for different types of attacks.

Knowledge for Security Testing

OWASP API Security Top 10 - 2023

- **API9:2023 - Improper Inventory Management**
  APIs tend to expose more endpoints than traditional web applications, making proper and updated documentation highly important. A proper inventory of hosts and deployed API versions also are important to mitigate issues such as deprecated API versions and exposed debug endpoints. Continue reading.

- **API10:2023 - Unsafe Consumption of APIs**
  Developers tend to trust data received from third-party APIs more than user input, and so tend to adopt weaker security standards. In order to compromise APIs, attackers go after integrated third-party services instead of trying to compromise the target API directly.

Knowledge for Security Testing

OWASP Top 10 – 2021

- **A01:2021-Broken Access Control**
  Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits

- **A02:2021-Cryptographic Failures**
  The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection.

Knowledge for Security Testing

OWASP Top 10 – 2021

- **A03:2021-Injection**
  An application is vulnerable to attack when User-supplied data is not validated, filtered, or sanitized by the application.
  Some of the more common injections are SQL, NoSQL, OS command, Object Relational Mapping (ORM), and LDAP injection.

- **A04:2021-Insecure Design**
  Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods. Threat modeling should be integrated into refinement sessions (or similar activities); look for changes in data flows and access control or other security controls.

Knowledge for Security Testing

OWASP Top 10 – 2021

- **A05:2021-Security Misconfiguration**
  The application might be vulnerable if the application is missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services. Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges). Or default accounts and their passwords are still enabled and unchanged.

- **A06:2021-Vulnerable and Outdated Components**
  You are likely vulnerable if you do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies.

Knowledge for Security Testing

OWASP Top 10 – 2021

- **A07:2021-Identification and Authentication Failures**
  Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. Authentication weaknesses include default, weak, or well-known passwords, such as "Password1" or "admin/admin" or permiting brute force or other automated attacks.

- **A08:2021-Software and Data Integrity Failures**
  Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs).

Knowledge for Security Testing

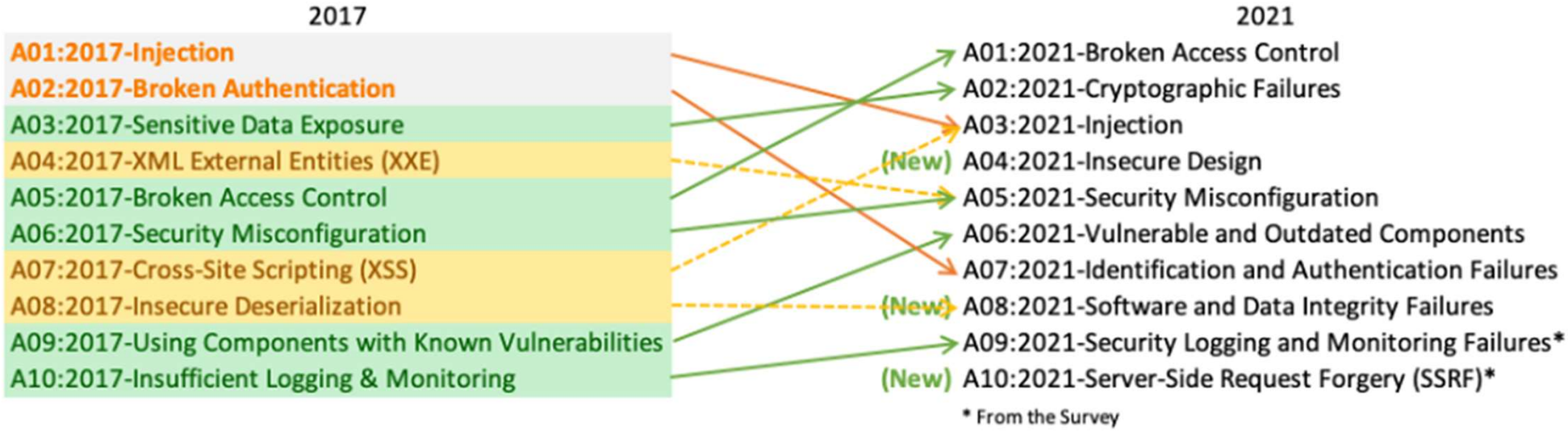
OWASP Top 10 – 2021

- **A09:2021-Security Logging and Monitoring Failures**
  This category is to help detect, escalate, and respond to active breaches. Without logging and monitoring, breaches cannot be detected. Insufficient logging, detection, monitoring, and active response occurs any time auditable events, such as logins, failed logins, and high-value transactions, are not logged.

- **A10:2021-Server-Side Request Forgery**
  SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list.

Knowledge for Security Testing

OWASP Top 10 – 2021



Knowledge for Security Testing

CWE/SANS Top 25 - 2023

1. Out-of-bounds Write In c you define an array to have the length 10, you can assign to index 11 an element -> you overwrite other data.
2. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
4. Use After Free You cleared memory, but keep the pointer ... dangerous.
5. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
6. Improper Input Validation
7. Out-of-bounds Read

Knowledge for Security Testing

CWE/SANS Top 25 - 2023

8. Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal’)
9. Cross-Site Request Forgery (CSRF)
10. Unrestricted Upload of File with Dangerous Type
11. Missing Authorization
12. NULL Pointer Dereference Static analysis useful.
13. Improper Authentication
14. Integer Overflow or Wraparound
15. Deserialization of Untrusted Data
16. Improper Neutralization of Special Elements used in a Command ('Command Injection')

Knowledge for Security Testing

CWE/SANS Top 25 - 2023

17. Improper Restriction of Operations within the Bounds of a Memory Buffer
18. Use of Hard-coded Credentials
19. Server-Side Request Forgery (SSRF)
20. Missing Authentication for Critical Function
21. Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
22. Improper Privilege Management
23. Improper Control of Generation of Code ('Code Injection')
24. Incorrect Authorization
25. Incorrect Default Permissions

Knowledge for Security Testing

FAQs

- Describe some common vulnerabilities listed in the **OWASP IoT, API Security,** or **Web Application Top 10** and **CWE/SANS Top 25** enumerations.

FH JOANNEUM
Elektronik und
Computer Engineering

References

- **OWASP Internet of Things**
  https://owasp.org/www-project-internet-of-things/
- **OWASP API Security Project**
  https://owasp.org/www-project-api-security/
- **Open Web Application Security Project (OWASP)**
  https://www.owasp.org/Top10/
- **Common Weakness Enumeration (CWE)**
  http://cwe.mitre.org/