2023

Security By Design

LEERDOEL ONDERZOEKSDOCUMENT

SEM PLASMEIJER

Contents

No table of contents entries found.

1. OWASP

1.1 Introduction

OWASP (Open Web Application Security Project) is a non-profit organisation with the focus on improving the safety and security of software and web applications. Their main function is compiling, ranking, and providing help for different security risks.

Every 3-4 years they publish a top 10 regarding that time periods most frequent, dangerous, or impactful security risks called the OWASP top 10. The most recent list contained the following items.

1. Broken Access Control
2. Cryptographic Failures
3. Injection / Cross-site Scripting (XSS)
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

Compared to the previous list made in 2017 none of the old top were removed but merged with other topics within the top 10. XSS previously ranked 7th in 2017 is now merged with other injection type security risks. The top 10 received 3 new entries in 2021 with Insecure Design, Data integrity failures and SSRF entering the top 10.

Beside ranking these risks OWASP gives solutions on how to prevent these different types of vulnerabilities. By giving examples of what might be labelled as wrong code and giving examples on how these can be circumvented.

2. Penetration Testing

2.1 Introduction

Security testing can be done in different ways, the most common type of testing is penetration testing or shortened to pentesting. Pentests are tests where usually a third party is asked to find vulnerabilities in a software solution with the goal of finding these risks before malicious actors can find these first. Pentests are in other words simulations of a bad actor trying to find a security.

Based on what kind of bad actor you might deal with different types of pentests can be run. For bad actors with no information at all the goals is to find any information to create larger attacks. Black box pentests are meant to simulate this event.

White box pentests are meant to simulate attacks from within an organisation. These tests will provide the bad actor with all information regarding test systems, network information, source code and details on what OS the server is running on.

Grey box tests are meant for simulating external bad actors. This test is meant to simulate what an attacker might achieve with leaked information regarding internal documentation or network information.

3. Design

Before designing a system security needs to be designed with it. Before starting on implementing business cases a systems designer might need to think out the following patterns.

- What rolls are contained within my system.
- What does each roll have access to in my system.
- How does my system secure and encrypt data.
- How does each service communicate securely with each other.

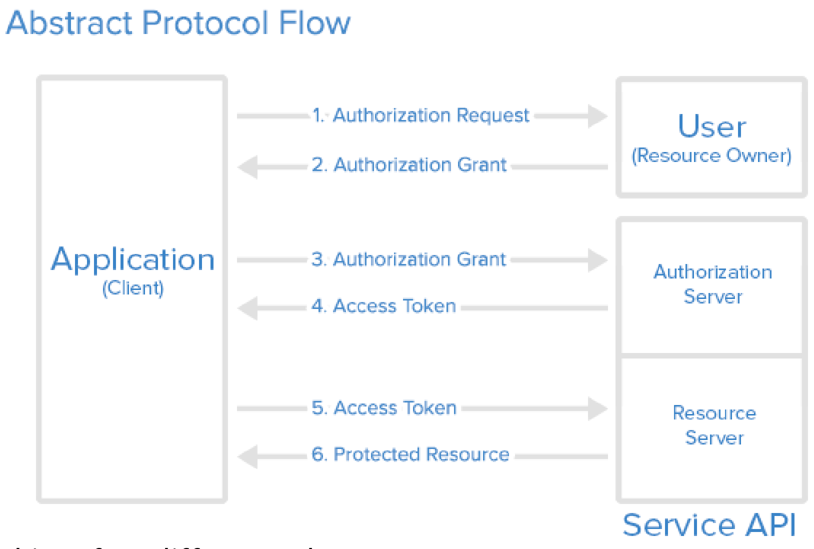
3.1 Layers of defence

Security of a system cannot be done with a single catch all file or management. Different types of risks require different types of security layers.

- Data Security
  - Identity and Access Management
  - Data Loss Prevention
  - Classification & Secure Archiving
  - File Encryption
- Network Security
  - Network Monitoring & Analysis
  - Sandboxing
  - Patch Management
- Cloud Security
  - Threat Detection
  - Threat Analysis
  - Threat Protection
  - Backup & Recovery

4. OAuth 2

OAuth 2 is security / authorization framework that allows applications to access third party limited access to a service using HTTP. The basic summary is that it creates and distributes user authentication on the service and allows a third-party or application to access that user account.



OAuth is divided into four different roles:

- **Resource Owner**: the resource owner has the job of authorizing who has access to the applications resources. Furthermore, they can set what scope of authorization a client has to the resource.
- **Client**: the client or application requests access to the resource. Before they can access any data, they must be authorized by the resource owner.
- **Resource Server**: The host of user accounts and who has protected access.
- **Authorization Server**: The authorization server verifies and grants access to the client through access tokens.

4.1 Route

In the image above is how OAuth usually interacts with authorization and how the different roles act upon each other.

In step 1 the application / client requests authorization rights to the access the resources on the server. As a response in step 2 the resource owner responds with a grant which the client can then use to identify itself to the authorization server and receive access to the resource server as a token. This token can be used to make HTTP calls to the Resource server in step 5 which then in step 6 gets confirmed by the server and replies with the requested resource.

4.2 Registration

To access the resource server an application will need to be registered with the application usually this happens through a registration form or a developer dashboard. During registration you will need to give the following 3 as a minimum:

- Application name
- Application Website URL
- Callback URL

The callback URL or Redirect URI is used to redirect back to the client application after requesting access token and authorization. The client will need to handle what interaction happens after the request gets authorized or denied.

4.2.1 Client

When registered a client id and client secret will be created. These credentials can be used to identify the application and redirect back to the client application.