

# **Web Application Security**

#### **Audience Course Web Application Security**

The course Web Application Security is intended for web developers who want to learn how to protect web applications against the many security risks.

#### **Prerequisites Course Web Application Security**

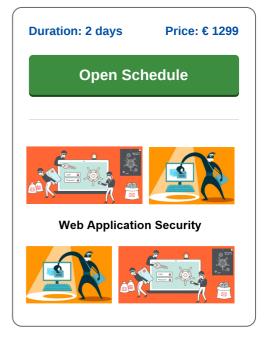
Experience in developing web applications is required to participate in this course. Experience with PHP or JavaScript is beneficial for understanding but not required.

#### **Realization Training Web Application Security**

The course Web Application Security has a hands-on character. The theory is treated on the basis of presentation slides and is interchanged with practical exercises. The course material is in English. Course times are from 9.30 up and to 16.30.

## **Certification Course Web Application Security**

After successful completion of the training the participants receive an official certificate Web Application Security.



## **Content Course Web Application Security**

The Web Application Security course discusses the most common security risks in web applications and how they can be tackled. At a time when attacks on applications seem to be on the rise, it is vital for developers to be aware of the types of threats and how the applications can be armed against them.

#### **Security Issues**

The course starts with a discussion of the most common security issues as identified in the Open Web Application Security Project (OWASP). This includes the risks of vulnerabilities in libraries, the importance of minimizing the attack surface of an application and vulnerabilities in authentication control.

#### **Cross Site Scripting**

In Cross-Site Scripting (XSS) attacks, malicious scripts are injected into a web site. Typically, this happens because the attacker makes JavaScript code run in the browser. XSS attacks are common and can occur anywhere in the application where user input is not validated.

## **SQL** Injection

Attention is also paid to SQL Injection, where an attacker places malicious code in SQL statements. SQL Injection is usually due to unchecked user input being used to create SQL statements. The consequences of SQL Injection can be serious such as data corruption, data theft or data destruction.

#### **Cross Site Request Forgery**

Next up in the course is the discussion of CSRF. Attention is paid to how CSRF executes malicious commands on behalf of a user trusted by the web application. Specially designed image tags or hidden forms are often used for this.

#### **Session Hijacking**

And Session Hijacking is on the program of the course as well. With Session Hijacking the attacker manages to obtain a session ID via sniffing techniques or XSS and then maliciously exploit it.

### **SSL Certificates**

Finally the course Web Application Security discusses securing web applications by means of SSL or TLS. An encrypted communication channel then ensures that data can be transported securely and digital certificates provide authentication.

Tel.: +31 (0) 30 - 737 0661

Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online



# **Modules Course Web Application Security**

Module 1 : Intro Security	Module 2 : Cross Site Scripting	Module 3 : SQL Injection
Security Risks	Malicious Code	SQL Injection Exploits
Top 10 OWASP Risks	Cookie Theft	Preventing SQL Injection
Sensitive Data Exposure	HTML Entity Encoding	Avoiding Dynamic Queries
Broken Authentication	XSS Prevention Rules	Prepared Statements
Social Engineering	Prevent Untrusted Data	Stored Procedures
Library Vulnerabilities	Attribute Encoding	Allow-List Input Validation
Sensitive Data Exposure	JavaScript Encoding	Escaping User Input
Attack Surface	HTML Encode JSON	Enforcing Least Privilege
Security Patches	CSS Encoding	Union Injections
Under Protected API's	URL Encoding	Database Differences
Coding for Security	Sanitize HTML Markup	Blind SQL Injection
Module 4 : Cross-Site Request Forgery	Module 5 : Session Hijacking	Module 6 : SSL Certificates
CSRF Attacks	Authentication Handshake	SSL and TLS
Malicious Requests	Session Cookies	Public and Private Keys
Stored CSRF Flaws	Cookie Theft	Encryption Methods
IMG or IFRAME Tags	Session Sidejacking	Asymmetric Encryption
Secret Cookies	Session Fixation	Symmetric Encryption
Only Accept POST	Man in the Middle	Hash Encryption
Form Tokens	Packet Sniffing	SSL Certificates
URL Rewriting	Hijack TCP-IP Session	Root Certificates
Same Origin Policy	Checking IP	Wildcard Certificates