**Secure Development for Clalit**

**Course 7564 – 40 Hours**



The Course will present security Guidelines, consideration and techniques for developing secure application, alongside explanations and demonstration of application specific attacks.   
Participant will learn and understand the different application security threats, and the right technique for mitigating each threat.

The course will contain demonstrations and exercises in python. However the course is not focused on a specific development or deployment environment, and all principles and concept demonstrated throughout the course are relevant to all applications.



* Hands-on developers and team leaders who wish to improve their security awareness and be certified for secure development in Clalit.
* System architects wishing to be able to assist the developers in creating a secure application



* Experience in developing Web Applications using modern programming languages (basic python is required)
* Recommended: Familiarity with the HTTP protocol
* Recommended: Familiarity with HTML
* Recommended: Familiarity with the SQL Language



Day 1

**OWASP Top 10 Application Security Risks and Mitigations**

* Injection
* Broken Authentication
* Sensitive Data Exposure
* XML External Entity (XXE) Vulnerabilities
* Broken Access Control
* Security Misconfiguration
* Cross Site Scripting (XSS)
* Insecure Deserialization
* Using Components with Known Vulnerabilities
* Insufficient Logging and Monitoring
* **Exercise: Web Hacking Challenge with BWAPP (requires Javascript, SQL, technical skills)**

**Input Validation and Output Sanitization**

* The Hacker Perspective
* Input Validation Goals
* Input Validation Strategies
* Implementing Input Validation
* Output Sanitization
* Exercise: Input Validation Attack and Defense (SQL Injection, XSS, requires Javascript and SQL)

Day 2

**Introduction to Information Security**

* Information Security Principles and CIA Triad
* Risk Analysis and Management
* Access Control (AAA)
* Getting started

**Cryptography Fundamentals**

* Cryptography Basic Concepts
* Symmetric Key Cryptography
* Asymmetric Key Cryptography
* Hashing Functions
* Public Key Infrastructure (PKI)
* Digital Signatures and Certificates
* Exercise: File Encryption and Hashing (requires python)

**Protect Data in Transit**

* Transport Security Overview
* SSL vs TLS
* The TLS Handshake
* Certificates
* SSL Offloading and SSL Termination
* HTTP Strict Transport Security (HSTS)
* Exercise:Intercepting HTTPS Traffic with Burp (requires installation and configuration)

**Protect Data at Rest**

* Database Security Overview
* Database Connection
* Managing Logins
* Storing Sensitive Data
* Database Configuration and Hardening

**Secrets Management**

* Understanding Application Secrets
* Safely Store Secrets in Development
* Protecting Production Secrets
* Exercise: Reading secrets from environment variables (requires python)

**Errors and Exceptions Handling**

* Exception Handling Overview
* Error Messages and Status Codes
* Global Error Handling

**Logging and Monitoring**

* Application Logs Overview
* What should and should not be logged
* Monitoring and Alerts

Day 3

**Identity and Access Management**

* What is Identity and Access Management (IAM)
* Authentication
* Authorization
* OAuth 2.0
* OpenID Connect (OIDC)
* JSON Web Token (JWT)
* JSON Object Signing and Encryption (JOSE)
* Bearer Tokens
* Exercise: Authentication with Bearer Tokens (requires python)

**Using Security Analysis Tools**

* Applying Static Code Analysis (SCA)
* Detecting Vulnerable Libraries
* Adding SCA and Vulnerable Library Detection to Build Pipelines

**Secure Application Development Processes**

* Security and Agile
* Secure DevOps (DevSecOps)
* Systems Development Life-Cycle (SDLC)
* **Introduction to Threat Modeling**
* What is Threat Modeling
* Threat Modeling Approaches
* Threat Modeling Methodologies
* Improve Application Security with Threat Modeling

Exercise: Threat modeling for a new feature

**Secure SDLC**

* **Principles of Secure Development**
* Security Design Principles
* OWASP Proactive Controls
* Exercise: Design a secure web application on paper

**Certification Exam**

The students will be tested on all the course material including the exercises.