Secure Software Development (Python)

Module 1: Introduction

Overview of Secure Software Development

Importance of security in software development

Key principles of secure coding

Overview of common security threats

Module 2: Basic Concepts in Secure Python Development

Python Programming Basics

Syntax and semantics

Data types and structures

Control flow statements

Functions and modules

Secure Coding Principles

Least privilege principle

Defense in depth

Fail securely

Avoiding security through obscurity

Module 3: Risk Identification in Python Applications

Common Security Risks in Python

Injection attacks (e.g., SQL Injection)

Cross-site Scripting (XSS)

Cross-site Request Forgery (CSRF)

Insecure deserialization

Tools and Techniques for Risk Identification

Static code analysis

Dynamic code analysis

Manual code reviews

Penetration testing

Module 4: Risk Mitigation Strategies

Input Validation and Sanitization

Importance of input validation

Techniques for validating and sanitizing input

Using libraries for input validation

Secure Authentication and Authorization

Implementing secure authentication mechanisms

Role-based access control (RBAC)

Token-based authentication

Data Protection

Encryption and hashing techniques

Secure storage of sensitive data

Managing secrets and keys

Error Handling and Logging

Best practices for error handling

Secure logging mechanisms

Avoiding information leakage

Module 5: Resource Scanning and Management

Dependency Management

Managing third-party libraries and dependencies

Identifying and mitigating vulnerabilities in dependencies

Using tools like pip-audit, safety, and Dependency-Check

Resource Scanning Tools

Static Application Security Testing (SAST) tools

Dynamic Application Security Testing (DAST) tools

Software Composition Analysis (SCA) tools

Module 6: Secure Software Development Lifecycle (SSDLC)

Introduction to SSDLC

Phases of SSDLC

Integrating security into each phase of SDLC

Security in DevOps (DevSecOps)

Incorporating security practices in CI/CD pipelines

Automated security testing

Module 7: Case Studies and Practical Applications

Analysis of Real-World Security Incidents

Case studies of notable security breaches

Lessons learned and how to prevent similar incidents

Hands-on Secure Coding Exercises

Practical exercises for identifying and fixing security issues

Building secure Python applications from scratch

Module 8: Final Project and Assessment

Final Project

Developing a secure Python application

Conducting a comprehensive security assessment of the project

Assessment and Certification

Evaluation of practical and theoretical knowledge

Certification of completion

Additional Resources

Reading Materials

Recommended books and articles on secure software development

Online Resources

Relevant websites, forums, and online courses

Tools and Software

List of useful tools for secure Python development