VERACODE

SECURING THE SOFTWARE THAT POWERS YOUR WORLD.

COURSE CATALOG

# eLearning Course Catalog

## Overview

Developers have to learn new languages, frameworks and skills throughout their careers, yet most never have the chance to learn to code securely. This becomes even more critical as development practices compress delivery schedules, putting pressure on the development team to solve its own problems without waiting on input from overtaxed security teams. All of these contributing factors cause security issues to be discovered later in the cycle, when they are more expensive to fix.

Veracode eLearning empowers developers, testers and security leads to develop secure applications, providing the critical skills they need to identify and address potential vulnerabilities before they hurt your bottom line. This turnkey training program can be rolled out across your organization without special hardware, software or travel to on-site training facilities. Veracode eLearning is delivered via the Veracode Application Security Platform, allowing you to reach development teams no matter where on earth they reside.

Veracode eLearning is comprised of two types of courses in distinct teaching styles to reach your users in the way that is best suited for them.

- **On-demand training courses** are integrated with the Veracode Application Security Platform and allows developers to learn when and where they need it.
- **Application Security (AppSec) Tutorials** offer refreshers and contextual recommendations to help developers fix vulnerabilities as they code. Development organizations that leverage Veracode eLearning see a 30 percent higher vulnerability fix rate.

Veracode eLearning offers relevant courses that help your organization meet their educational and career development goals. Students who complete the coursework are eligible for continuing professional education (CPE) credits as part of $(ISC)^2$, CISSP, and CSSLP certifications. Additionally there are a variety of courses that Veracode offers that meet the PCI DSS requirement 6.5 for the training of developers in secure coding techniques.

This document provides detailed descriptions of all the online secure development training courses that that are included with Veracode eLearning. The courses have been categorized into skill levels ranging from Introductory to Expert, and were designed to teach all experience levels of individuals in your organization about the importance of secure practices during the software development cycle.

| Course Title | Duration | Intended Audience |
|---|---|---|
| **Introductory Security Courses** | | |
| General Security Awareness | 1 hour | All employees and contractors |
| Introduction to Web Application Security | 1 hour | Security Professionals, Software Developers, Project Managers, Quality Assurance Staff |
| Overview of Mobile Application Security | 30 minutes | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| Veracode Application Security Fundamentals Assessment | 30 minutes | Exam intended to baseline the application security proficiency of your technology community. |
| **Intermediate Security Courses** | | |
| Application Security Testing | 1 hour | Security Professionals and Software Developers |
| Introduction to Payment Card Industry (PCI) Data Security Standard (DSS) 3.2 for Developers | 1 hour | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| Secure Software Remediation Basics | 1 hour | Security Professionals, Software Developers and Software Quality Assurance Staff |
| **Advanced Security Courses** | | |
| Authentication & Authorization for Android | 30 minutes | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| Authentication & Authorization for iOS | 30 minutes | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| C / C++ Memory Management Risks and Best Practices | 1 hour | Software Developers |
| **Expert Security Courses** | | |
| Cross-Site Request Forgery (CSRF) Explained | 20 minutes | Security Professionals and Software Developers |
| Data Protection for Android | 30 minutes | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| Secure Architecture & Design | 1 hour | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| Threat Modeling | 1 hour | Security Professionals and Software Developers |
| Validation and Encoding for Android | 30 minutes | Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers |
| **Secure Coding for Java with OWASP Top Ten** | | |
| Secure Coding for Java - Authentication | 30 minutes | Software Developers |
| Secure Coding for Java - Authorization | 15 minutes | Software Developers |
| Secure Coding for Java - Configuration and Deployment | 45 minutes | Software Developers |
| Secure Coding for Java - Data Protecting | 30 minutes | Software Developers |

| Course Title | Duration | Intended Audience |
|---|---|---|
| Secure Coding for Java - Information Handling, Non-Repudiation and Auditing | 45 minutes | Software Developers |
| Secure Coding for Java - Trust Boundaries | 15 minutes | Software Developers |
| Secure Coding for Java - Validation and Encoding | 30 minutes | Software Developers |
| **Secure Coding for .NET with OWASP Top Ten** | | |
| Secure Coding for .NET - Authentication | 30 minutes | Software Developers |
| Secure Coding for .NET - Authorization | 15 minutes | Software Developers |
| Secure Coding for .NET - Configuration and Deployment | 45 minutes | Software Developers |
| Secure Coding for .NET - Data Protecting | 30 minutes | Software Developers |
| Secure Coding for .NET - Information Handling, Non-Repudiation and Auditing | 45 minutes | Software Developers |
| Secure Coding for .NET - Trust Boundaries | 15 minutes | Software Developers |
| Secure Coding for .NET - Validation and Encoding | 30 minutes | Software Developers |
| **Secure Coding for PHP with OWASP Top Ten** | | |
| Secure Coding for PHP - Authentication | 30 minutes | Software Developers |
| Secure Coding for PHP - Authorization | 15 minutes | Software Developers |
| Secure Coding for PHP - Configuration and Deployment | 45 minutes | Software Developers |
| Secure Coding for PHP - Data Protecting | 30 minutes | Software Developers |
| Secure Coding for PHP - Information Handling, Non-Repudiation and Auditing | 45 minutes | Software Developers |
| Secure Coding for PHP - Trust Boundaries | 15 minutes | Software Developers |
| Secure Coding for PHP -Validation and Encoding | 30 minutes | Software Developers |
| **AppSec Tutorials** | | |
| AppSec Tutorial - CRLF Injection | 15 minutes | Software Developers |
| AppSec Tutorial - Cross Site Scripting (XSS) | 10 minutes | Software Developers |
| AppSec Tutorial - Directory Traversal | 10 minutes | Software Developers |
| AppSec Tutorial - Operating System Command Injection | 15 minutes | Software Developers |
| AppSec Tutorial - SQL Injection | 10 minutes | Software Developers |

## General Security Awareness

Most security professionals and software developers are aware of security concerns within the organization, but what about the rest of your staff? The purpose of security awareness training is to help computer users make smart decisions regarding the security choices that they face every day at work. This self-paced, e-Learning course provides an understanding of the basics of security awareness for all employees and contractors, not just technical staff. This course teaches computer users about common security threats to an organization, how to make informed security decisions, and ways in which staff and contractors can become an organization's first line of defense against external and internal security breaches.

**Course Duration:** 1 hour

**Prerequisites:** No previous information security experience is required. Basic computer experience is required

**Intended Audience:** All employees and contractors

### Lesson 1: Why is Security Important?

**Objectives:** After completing this lesson, you should be able to:

- Demonstrate an understanding of security as it relates to workplace information systems
- Demonstrate knowledge of the expanding nature of security threats in the workplace

### Lesson 2: Security in the Workplace

**Objectives:** After completing this lesson, you should be able to:

- Describe best practices for avoiding social engineering attacks
- Describe best practices regarding the use of email and electronic communications, including avoiding malware infections

### Lesson 3: Security Best Practices

**Objectives:** After completing this lesson, you should be able to:

- Describe best practices in password use and strength
- Demonstrate knowledge of the security implications of remote access to the workplace and the use of mobile devices

## Introduction to Web Application Security

This self-paced, e-Learning course provides students with the basic concepts and terminology for understanding application security issues. It provides a definition of application-level security and demonstrates how these concerns extend beyond those of traditional infrastructure security. It also provides an explanation of common application security vulnerabilities such as SQL injection, Cross Site Scripting (XSS) and authorization issues. Armed with this knowledge, developers, QA testers and security personnel can understand and start to be able to address application-level threats.

**Course Duration:** 1 hour

**Intended Audience:** Security Professionals, Developers, Project Managers, Quality Assurance Staff

### Lesson 1: Intro & Concepts

**Objectives:** After completing this lesson, you should be able to:

- Explain how intended application functionality differs from the intended functionality and how it is interesting to an attacker
- Realize the potential for application inputs to be used as avenues for attack

### Lesson 2: Real Case Studies - Notable Breaches

**Objectives:** After completing this lesson, you should be able to:

- Appreciate the impact of poor security in production environments
- Justify the mitigation effort to minimize exposed attack surfaces

### Lesson 3: Application Attack Demonstration

**Objectives:** After completing this lesson, you should be able to:

- Understand the approaches an attacker uses to find application-level vulnerabilities
- Understand the potential for malicious use of features in a vulnerable application

### Lesson 4: What is Application Security and Why is it Important?

**Objectives:** After completing this lesson, you should be able to:

- Provide a working definition of application security
- Provide explanations of the chief application security concerns: Confidentiality, Integrity and Availability
- Explain why application security is important for organizations to address
- Describe the roles that major regulatory requirements play in secure application development

## Lesson 5: SQL Injection Activity

**Objectives:** After completing this lesson, you should be able to:

- Understand the basics of an SQL Injection attack
- Understand the potential impact of exploited SQL injection vulnerabilities
- Understand the basics of protecting an application from injection attacks

## Lesson 6: HTTP Basics

**Objectives:** After completing this lesson, you should be able to:

- Explain the difference between GET and POST requests
- Explain the Lifecycle of HTTP Requests
- Explain the benefits and risks of session authentication over HTTP Basic authentication

## Lesson 7: Cross-Site Scripting Activity

**Objectives:** After completing this lesson, you should be able to:

- Describe the mechanics behind Cross-Site Scripting (XSS) vulnerabilities and attacks
- Understand how XSS can abuse a user's trust
- Understand the types of risks the exploitation of XSS vulnerabilities poses to web applications

## Overview of Mobile Application Security

Overview of Mobile Application Security gives a step-by-step guide on how to build a basic threat model for a smartphone application. This threat model is then used as a framework for making better decisions about how to design and build applications as well as how to test the security of existing applications. By understanding how mobile applications are connected to other systems, developers will understand how mobile applications can be vulnerable and sensitive data exposed. This self-paced, e-Learning course provides an introduction to the basic concepts and best practices of secure development for mobile devices, concentrating on Android and iOS.

**Course Duration:** 30 minutes

**Intended Audience:** Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers

### Lesson 1: Overview of Mobile Application Security

**Objectives:** After completing this lesson, you should be able to:

- Understand the mobile universe and the capabilities of mobile devices
- Explain how mobile applications pose different security risks from web applications

### Lesson 2: The Mobile Application Threat Model

**Objectives:** After completing this lesson, you should be able to:

- Explain a threat model for mobile applications
- Articulate the risks of a mobile application and how mobile developers can strike a balance between functionality and security

### Lesson 3: Threats Facing Mobile Applications

**Objectives:** After completing this lesson, you should be able to:

- Understand the threats inherent in mobile applications
- Understand the capabilities of different development platforms (iOS and Android)
- Understand how to take advantage of mobile capabilities without exposing users to unnecessary risks

## Application Security Testing

Application security testing is a way for organizations to identify and mitigate security vulnerabilities in their applications. This course covers the general approach used in a security assessment; the lessons identify the steps that take place and the activities that are performed during an assessment. The course covers the tools and techniques that are used to identify and follow-up on vulnerabilities discovered during the baseline and targeted testing steps of an assessment; the following assessment activities are explained: static analysis, dynamic analysis, forensic analysis, penetration testing, and code review. The lessons also describe how to rate vulnerabilities observed during an assessment according to the DREAD rating system and how to explain remediation recommendations in an assessment report.

**Course Duration:** 1-hour

**Prerequisites:** Introduction to Web Application Security and Secure Coding for Java/.Net

**Intended Audience:** Security Professionals, Software Developers, Mobile Application Developers, Penetration Testers

### Lesson 1: General Assessment Approach

**Objectives:** After completing this lesson, you should be able to:

- Identify the steps that take place during a security assessment and explain the purpose of each step
  - Assessment Preparation
  - Baseline Review and Testing
  - Threat Modeling
  - Targeted Testing
  - Reporting
- Understand the differences between static, dynamic and forensic analysis
- Describe the code review process and how to investigate observed vulnerabilities
- Understand when to use automated versus manual testing to find different types of security flaws

### Lesson 2: Assessment Preparation

**Objectives:** After completing this lesson, you should be able to:

- Describe a methodology for prioritizing applications to be assessed
- Describe decision criteria for selecting different assessment activities for an application
- Explain the resources needed to prepare for a dynamic application assessment
- Explain the resources needed to prepare for a static application assessment

## Lesson 3: Threat Modeling

**Objectives:** After completing this lesson, you should be able to:

- Demonstrate the process of decomposing a system into a set of assets and the data flows between them
- Demonstrate the process of applying STRIDE to a data flow diagram to identify threats

## Lesson 4: Baseline Review and Testing

**Objectives:** After completing this lesson, you should be able to:

- Describe an approach for manually reviewing and verifying dynamic scan results
- Describe an approach for manually reviewing and verifying static scan results
- Describe the OWASP Application Security Verification Standard (ASVS)

## Lesson 5: Targeted Testing

**Objectives:** After completing this lesson, you should be able to:

- Explain how threats developed earlier in the review process should be translated to targeted testing activities
- Provide an overview of manual code review techniques

## Lesson 6: Reporting

**Objectives:** After completing this lesson, you should be able to:

- Understand how to use different approaches and systems to rate application vulnerabilities
- Understand how to classify and explain a rating in a security assessment
- Understand how to report mitigation steps, risks and remediation recommendations for a vulnerability
- Learn how to find additional information about known security vulnerabilities online

## Introduction to Payment Card Industry (PCI) Data Security Standard (DSS) 3.2 for Developers

This course will expose application developers to the Payment Card Industry (PCI) Data Security Standard (DSS). The course will provide background and context on why organizations must be PCI compliant, the risk of and penalties for non-compliance, and developer responsibilities associated with PCI DSS requirements. The course will include examples of threats to PCI cardholder data, an overview of PCI-relevant secure coding practices, and methods to maintain PCI compliance over an extended period. This course is designed for developers at all levels of experience and is programming language agnostic. Upon successful completion of this course, students should be able to discuss PCI DSS requirements, apply relevant knowledge to their roles, and be able to demonstrate to PCI assessors that they have completed a basic overview of PCI DSS and secure coding techniques.

**Course Duration:** 1 hour

**Intended Audience:** Software Developers, Mobile Application Developers, Software Development Managers

### Lesson 1: Overview of PCI DSS Requirements and Compliance

**Objectives:** After completing this lesson, you should be able to:

- Describe the 12 requirements of PCI DSS
- Describe the history of PCI DSS
- Understand the risk of non-compliance

### Lesson 2: The Importance of PCI DSS in the Marketplace

**Objectives: After completing this lesson, you should be able to:**

- Describe the purpose of PCI DSS
- Debunk common myths of PCI DSS
- Identify threats to cardholder data

### Lesson 3: Requirement 6 In-Depth

**Objectives: After completing this lesson, you should be able to:**

- Comply with PCI DSS requirements to develop and maintain secure systems and applications
- Be familiar with secure system management practices such as:
  - Patch management
  - Change control
  - Code review
  - Vulnerability scanning
- Be familiar with secure software development practices such as:
  - Input validation
  - Output encoding
  - Secure data storage

## Secure Software Remediation Basics

The security industry often pays a tremendous amount of attention to finding security vulnerabilities. This is done via code review, penetration testing and other assessment methods. Unfortunately, finding vulnerabilities is only the first step toward actually addressing the associated risks, and addressing these risks is arguably the most critical step in the vulnerability management process. Complicating matters is the fact that most application security vulnerabilities cannot be fixed by members of the security team because they require code-level changes in order to address the underlying issue successfully. Therefore, security vulnerabilities need to be communicated and transferred to software development teams and then prioritized and added to their workloads. This self-paced, e-Learning course examines steps required to remediate software-level vulnerabilities properly, and recommends best practices organizations can use to be successful in their remediation efforts.

**Course Duration:** 1 hour

**Intended Audience:** Security Professionals, Developers and Software Quality Assurance Staff

### Lesson 1: Software Security Remediation Basics

**Objectives:** After completing this lesson, you should be able to:

- Understand the overall purpose, process, impact and phases of software security remediation projects

### Lesson 2: Phase One – Inception

**Objectives:** After completing this lesson, you should be able to:

- Identify individuals and teams that should be involved in software security remediation projects
- Understand how to create a successful timeline and budget

### Lesson 3: Phase Two – Planning

**Objectives:** After completing this lesson, you should be able to:

- Understand risks associated with software vulnerabilities and how risk is calculated
- Explain how manual and automated testing is used to find and confirm vulnerabilities
- Calculate the level of effort needed from various teams
- Schedule a software security remediation project

### Lesson 4: Phase Three – Execution

**Objectives:** After completing this lesson, you should be able to:

- Explain the steps and methods necessary to fix vulnerabilities
- Understand how to test the quality of vulnerability fixes
- Provide metrics used to evaluate a software security remediation project

## Authentication & Authorization for Android

Authentication and Authorization are the first line of defense in securing a mobile application, but they are not fool-proof. Developers need to understand the risks of these techniques, and how to protect against these risks. This course, for Android, covers industry best practices for protecting a mobile application from malicious users using these methods. This self-paced, e-Learning course provides an overview of common authentication and authorization approaches for the Android platforms.

**Course Duration:** 30 minutes

**Intended Audience:** Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers

### Lesson 1: Authentication and Authorization

**Objectives:** After completing this lesson, you should be able to:

- Define authentication and authorization
- Describe session management for the platform

### Lesson 2: Lack of Data Protection In-Transit

**Objectives:** After completing this lesson, you should be able to:

- Explain various scenarios of how data can be exploited in transit
- Understand how to protect data in transit for the platform

### Lesson 3: Failure to Protect Resources with Strong Authentication

**Objectives:** After completing this lesson, you should be able to:

- Explain how authentication can be exploitable
- Describe authentication schemes can be enhanced on the platform

### Lesson 4: Insecure On-Device Credential Storage

**Objectives:** After completing this lesson, you should be able to:

- Describe the types of information that can be gleaned from a mobile device
- Explain the best practices for secure data storage

## Authentication & Authorization for iOS

Authentication and Authorization are the first line of defense in securing a mobile application, but they are not fool-proof. Developers need to understand the risks of these techniques, and how to protect against these risks. This course, for iOS, covers industry best practices for protecting a mobile application from malicious users using these methods. This self-paced, e-Learning course provides an overview of common authentication and authorization approaches for the iOS platforms.

**Course Duration:** 30 minutes

**Intended Audience:** Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers

### Lesson 1: Authentication and Authorization

**Objectives:** After completing this lesson, you should be able to:

- Define authentication and authorization
- Describe session management for the platform

### Lesson 2: Lack of Data Protection In-Transit

**Objectives:** After completing this lesson, you should be able to:

- Explain various scenarios of how data can be exploited in transit
- Understand how to protect data in transit for the platform

### Lesson 3: Failure to Protect Resources with Strong Authentication

**Objectives:** After completing this lesson, you should be able to:

- Explain how authentication can be exploitable
- Describe authentication schemes can be enhanced on the platform

### Lesson 4: Insecure On-Device Credential Storage

**Objectives:** After completing this lesson, you should be able to:

- Describe the types of information that can be gleaned from a mobile device
- Explain the best practices for secure data storage

## C / C++ Memory Management Risks and Best Practices

C and C++ are widely-adopted, deeply influential, and supported by a tremendous variety of frameworks and development environments. This speaks to the diversity of C/C++ developers and applications. It should also remind developers that C/C++ security risks and exploits are well-known among attackers.

Memory management is the most well-known risk with C/C++, and for good reason. This course will cover memory management fundamentals and common coding flaws that open an application to buffer overflow exploits and other attacks. The course will cover secure coding practices throughout, providing fixes to coding flaws as well as recommendations for comprehensive memory management solutions.

**Course Duration:** 1-hour

**Intended Audience:** Developers

### Lesson 1: Overview of C / C++ Memory Management

**Objectives:** After completing this lesson, you should be able to:

- Describe the characteristics of C/C++ memory management that present security risks

### Lesson 2: Stack and Heap Architecture

**Objectives:** After completing this lesson, you should be able to:

- Recognize stack and heap overflow
- Identify how these vulnerabilities can be exploited and avoided

### Lesson 3: Common Coding Flaws

**Objectives:** After completing this lesson, you should be able to:

- Recognize the following common coding pitfalls and best practices to avoid them:
  - Unchecked array indexing and reference
  - Format strings
  - Integer overflows
  - Double free
  - Improper bounds checking
  - Memory footprints of different data types
  - Off-by-one errors
  - Signed and unsigned integers
  - Memory leaks

### Lesson 4: Memory Management Solutions

**Objectives:** After completing this lesson, you should be able to:

- Recognize other coding solutions that can aid in memory management

## Secure Coding for Java with OWASP Top Ten

Seven related courses where developers will learn application security practices, associated vulnerabilities including the OWASP Top Ten and secure coding techniques in Java. Each course trains on a specific topic. Courses can be taken in sequence or individually at the developers own pace and can be used as a reference for developers. Each course includes multiple knowledge checks and a final quiz to assess the learner's understanding of the topic.

**Course Duration** (7 courses)**:** 210 minutes

**Intended Audience:** Developers

### Secure Coding for Java with OWASP Top Ten - Trust Boundaries

**Course Duration:** 15 minutes

**Course Summary:** This course educates learners on identifying trust boundaries, how attackers can exploit applications and how to properly handle trust boundaries during design and development.

**Objectives:** After completing this course, a user will be able to:

- Describe the concept of trust boundaries and how they apply to application security
- Demonstrate an understanding of general approaches for handling trust boundaries in applications

### Secure Coding for Java with OWASP Top Ten - Authentication

**Course Duration:** 30 minutes

**Course Summary:** This course educates the learner on authentication solutions, common vulnerabilities and resources available to implement authentication in an application.

**Objectives:** After completing this course, a user will be able to:

- Identify common authentication approaches
- Identify common authentication vulnerabilities

### Secure Coding for Java with OWASP Top Ten - Authorization

**Course Duration:** 15 minutes

**Course Summary:** This course educates the learner on access controls, possible exploits and ways to implement authorization.

**Objectives:** After completing this course, a user will be able to:

Describe common approaches for authorizing system access

- Describe where authorization should occur
- Demonstrate knowledge of common authorization vulnerabilities

## Secure Coding for Java with OWSAP Top Ten - Validation and Encoding

**Course Duration:** 30 minutes

**Course Summary:** This course educates the user on input validation and how attackers utilize flaws such SQL Injection or Cross Site Scripting (XSS) to compromise an application. This course provides best practices for implementing validation.

**Objectives:** After completing this course, a user will be able to:

Describe best practices for input validation

- Identify common vulnerabilities that proper validation can help address

## Secure Coding for Java with OWSAP Top Ten - Information Handling, Non-Repudiation and Auditing

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on ways that information an application provides to users can be exploited by attackers. The course gives practical ways to design an application to log information for auditing porpoises as well as report errors without providing sensitive information that could comprise the application.

**Objectives:** After completing this course, a user will be able to:

- Describe the risks associated with poor information and error handling
- Describe best practices for containing sensitive information and handling application failure
- Describe the value of non-repudiation, separation of duties, and support for auditing
- Identify best practices for logging and reporting error conditions

## Secure Coding for Java with OWASP Top Ten - Data Protection

**Course Duration:** 30 minutes

**Course Summary:** This course covers the importance of protecting data in motion as well as at rest as well as the cryptographic algorithms that applications can implement to protect data.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of the general concepts of modern cryptography
- Describe cryptographic best practices and common mistakes
- Identify approaches for handling data classification standards

## Secure Coding for Java with OWASP Top Ten - Configuration and Deployment

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on proper configuration and deployment to prevent common vulnerabilities such as SQL Injection or Cross Site Scripting (XSS) to compromise an application..  The principle of least privilege will also be discussed and how it should be applied to application design, configuration, and deployment.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of how proper configuration and deployment can manage the impact of existing vulnerabilities and prevent others
- Describe common configuration and deployment flaws and the danger they post to applications

## Secure Coding for .NET with OWASP Top Ten

Seven related courses where developers will learn application security practices, associated vulnerabilities including the OWASP Top Ten and secure coding techniques in .NET. Each course trains on a specific topic. Courses can be taken in sequence or individually at the developers own pace and can be used as a reference for developers during coding and remediation. Each course include multiple knowledge checks and a final quiz to assess the learner's understanding of the topic.

**Total Duration** (7 Courses): 210 Minutes

**Intended Audience:** Developers

### Secure Coding for .Net with OWASP Top Ten - Authentication

**Course Duration:** 30 minutes

**Course Summary:** This course educates the leaner on authentication solutions, common vulnerabilities and resources available to implement authentication in an application.

**Objectives:** After completing this course, a user will be able to:

- Identify common authentication approaches
- Identify common authentication vulnerabilities

### Secure Coding for .NET with OWASP Top Ten - Authorization

**Course Duration:** 15 minutes

**Course Summary:** This course educates the learner on access controls, possible exploits and ways to implement authorization.

**Objectives:** After completing this course, a user will be able to:

- Describe common approaches for authorizing system access
- Describe where authorization should occur
- Demonstrate knowledge of common authorization vulnerabilities

### Secure Coding for .NET with OWASP Top Ten - Configuration and Deployment

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on proper configuration and deployment to prevent common vulnerabilities. The principle of least privilege will also be discussed and how it should be applied to application design, configuration, and deployment.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of how proper configuration and deployment can manage the impact of existing vulnerabilities and prevent others

- Describe common configuration and deployment flaws and the danger they post to applications

## Secure Coding for .NET with OWASP Top Ten - Data Protection

**Course Duration:** 30 minutes

**Course Summary:** This course covers the importance of protecting data in motion as well as at rest as well as the cryptographic algorithms that applications can implement to protect data.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of the general concepts of modern cryptography
- Describe cryptographic best practices and common mistakes
- Identify approaches for handling data classification standards

## Secure Coding for .NET with OWASP Top Ten - Information and Error Handling

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on ways that information an application provides to users can be exploited by attackers. The course gives practical ways to design an application to log information for auditing porpoises as well as report errors without providing sensitive information that could comprise the application.

**Objectives:** After completing this course, a user will be able to:

- Describe the risks associated with poor information and error handling
- Describe best practices for containing sensitive information and handling application failure
- Describe the value of non-repudiation, separation of duties, and support for auditing
- Identify best practices for logging and reporting error conditions

## Secure Coding for .NET with OWASP Top Ten - Trust Boundaries

**Course Duration:** 15 minutes

**Course Summary:** This course educates learners on identifying trust boundaries, how attackers can exploit applications and how to properly handle trust boundaries during design and development.

**Objectives:** After completing this course, a user will be able to:

- Describe the concept of trust boundaries and how they apply to application security
- Demonstrate an understanding of general approaches for handling trust boundaries in applications

## Secure Coding for .NET with OWASP Top Ten - Validation and Encoding

**Course Duration:** 30 minutes

**Course Summary:** This course educates the user on proper configuration and deployment to prevent common vulnerabilities such as SQL Injection or Cross Site Scripting (XSS) to compromise an application..  The principle of least privilege will also be discussed and how it should be applied to application design, configuration, and deployment.

**Objectives:** After completing this course, a user will be able to:

- Describe best practices for input validation
- Identify common vulnerabilities that proper validation can help address

## Secure Coding for PHP with OWASP Top Ten

Seven related courses where developers will learn application security practices, associated vulnerabilities including the OWASP Top Ten and secure coding techniques in PHP. Each course trains on a specific topic. Courses can be taken in sequence or individually at the developers own pace and can be used as a reference for developers during coding and remediation. Each course include multiple knowledge checks and a final quiz to assess the learner's understanding of the topic.

**Total Duration** (7 Courses): 210 Minutes

**Intended Audience:** Developers

### Secure Coding for PHP with OWASP Top Ten - Authentication

**Course Duration:** 30 minutes

**Course Summary:** This course educates the leaner on authentication solutions, common vulnerabilities and resources available to implement authentication in a PHP application.

**Objectives:** After completing this course, a user will be able to:

- Identify common authentication approaches
- Identify common authentication vulnerabilities

### Secure Coding for PHP with OWASP Top Ten - Authorization

**Course Duration:** 15 minutes

**Course Summary:** This course educates the learner on access controls, possible exploits and ways to implement authorization for a PHP application.

**Objectives:** After completing this course, a user will be able to:

- Describe common approaches for authorizing system access
- Describe where authorization should occur
- Demonstrate knowledge of common authorization vulnerabilities

### Secure Coding for PHP with OWASP Top Ten - Configuration and Deployment

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on proper configuration and deployment to prevent common vulnerabilities. The principle of least privilege will also be discussed and how it should be applied to application design, configuration, and deployment.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of how proper configuration and deployment can manage the impact of existing vulnerabilities and prevent others

- Describe common configuration and deployment flaws and the danger they post to applications

## Secure Coding for PHP with OWASP Top Ten - Data Protection

**Course Duration:** 30 minutes

**Course Summary:** This course covers the importance of protecting data in motion as well as at rest as well as the cryptographic algorithms that applications can implement to protect data.

**Objectives:** After completing this course, a user will be able to:

- Demonstrate knowledge of the general concepts of modern cryptography
- Describe cryptographic best practices and common mistakes
- Identify approaches for handling data classification standards

## Secure Coding for PHP with OWASP Top Ten - Information and Error Handling

**Course Duration:** 45 minutes

**Course Summary:** This course educates the user on ways that information an application provides to users can be exploited by attackers. The course gives practical ways to design an application to log information for auditing porpoises as well as report errors without providing sensitive information that could comprise the application.

**Objectives:** After completing this course, a user will be able to:

- Describe the risks associated with poor information and error handling
- Describe best practices for containing sensitive information and handling application failure
- Describe the value of non-repudiation, separation of duties, and support for auditing
- Identify best practices for logging and reporting error conditions

## Secure Coding for PHP with OWASP Top Ten - Trust Boundaries

**Course Duration:** 15 minutes

**Course Summary:** This course educates learners on identifying trust boundaries, how attackers can exploit applications and how to properly handle trust boundaries during design and development.

**Objectives:** After completing this course, a user will be able to:

- Describe the concept of trust boundaries and how they apply to application security
- Demonstrate an understanding of general approaches for handling trust boundaries in applications

## Secure Coding for PHP with OWASP Top Ten - Validation and Encoding

**Course Duration:** 30 minutes

**Course Summary:** This course educates the user on proper configuration and deployment to prevent common vulnerabilities such as SQL Injection or Cross Site Scripting (XSS) to compromise an application..  The principle of least privilege will also be discussed and how it should be applied to application design, configuration, and deployment.

**Objectives:** After completing this course, a user will be able to:

- Describe best practices for input validation
- Identify common vulnerabilities that proper validation can help address

## Cross Site Request Forgery (CSRF) Explained

Cross-Site Request Forgery (CSRF) is a serious and often-misunderstood web application vulnerability. This self-paced, e-Learning course goes into detail about the anatomy of a CSRF vulnerability as well as how security analysts can identify CSRF vulnerabilities and how developers can design and build applications resistant to CSRF attacks. Interactive examples and videos demonstrate the subtleties of CSRF vulnerabilities and how malicious attackers exploit them.

**Course Duration:** 20 minutes

**Intended Audience:** Security Professionals and Developers

### Lesson 1: Cross-Site Request Forgery (CSRF) Vulnerabilities

**Objectives:** After completing this lesson, you should be able to:

- Understand what a CSRF vulnerability is
- Test applications to identify potential CSRF vulnerabilities
- Build applications free from CSRF vulnerabilities

## Data Protection for Android

The Android platform has specific facilities for storing and transmitting data. Some security restrictions are built-in to the Android platform, but developers need to take extra steps to ensure more secure protection. This course covers best practices for protecting data on the Android platform. This self-paced, e-Learning course provides an overview of techniques for data protection on the Android platform.

**Course Duration:** 30 minutes

**Intended Audience:** Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers

### Lesson 1: Securing Stored Data

**Objectives:** After completing this lesson, you should be able to:

- Describe how the Android platform stores files, data bases and shared preferences
- Explain Android encryption methods and best practices

### Lesson 2: Securing Data in Transit

**Objectives:** After completing this lesson, you should be able to:

- Enumerate the benefits of SSL
- Discuss methods to avoid in order to secure data in transit on the Android platform

## Secure Architecture & Design

Security testing and remediation are important in a software project to protect the organization; however, these measures are reactive and can be costly. This self-paced, e-Learning course covers the general concepts and approach to designing secure software architecture from the ground-up. We will discuss finding appropriate solutions for functional security requirements such as authentication, access control, and secure storage. The course also explains how to analyze the architecture for business policy needs and risks from external dependencies. The final section of this course discusses data flow and control flow analysis – data flow diagrams and control flow graphs are explained and utilized.

**Course Duration:** 1-hour

**Prerequisites:** Introduction to Web Application Security

**Intended Audience:** Software Developers, Mobile Application Developers

### Lesson 1: Secure Design – Functional Security Requirements and Solutions

**Objectives:** After completing this lesson, you should be able to:

- Identify the potential risks, requirements, and solutions for each of the functional security domains:
  - Authentication and Session Management
  - Access Control
  - Input Validation and Output Encoding
  - Cryptography and Data Protection
  - Error Handling and Logging
  - Communication and HTTP Security
  - Files and Resources

### Lesson 2: Secure Design – Use and Abuse Cases

**Objectives:** After completing this lesson, you should be able to:

- Understand how to identify application use and abuse cases and how they are used
- Explain how to create a diagram mapping abuse cases to use cases to identify interactions between the system, users, attackers, and security controls

## Lesson 3: Secure Architecture – Business Controls and Risks from Dependencies

**Objectives:** After completing this lesson, you should be able to:

- Understand how to document and analyze the application's architecture through the use of the following tools:
  - External Dependencies
  - Entry Points
  - Assets
  - Trust Levels
- Explain the need for business controls to protect an organization's systems and assets
- Identify and analyze the risks from the application's infrastructure (platform, frameworks, and system components)

## Lesson 4: Secure Architecture – Data Flow and Control Flow Analysis

**Objectives:** After completing this lesson, you should be able to:

- Understand how to create and analyze Data Flow Diagrams (DFDs) and Control Flow Graphs (CFGs)
- Utilize the steps of the STRIDE approach to categorize threats and determine appropriate countermeasures based on the STRIDE category:
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege

## Threat Modeling

Threat Modeling is a key practice for organizations wanting to design and develop secure applications as it helps to identify potential security vulnerabilities early in the process when they are inexpensive to fix. This self-paced, e-Learning course walks through the Threat Modeling process step by step so that students understand the value of Threat Modeling and can build threat models for their own systems.

**Course Duration:** 1 hour

**Intended Audience:** Security Professionals and Developers

### Lesson 1: Threat Modeling: Principles and Practices

**Objectives:** After completing this lesson, you should be able to:

- Understand what Threat Modeling is
- Identify when it is appropriate to use
- Explain why Threat Modeling is useful
- Understand how to use Threat Modeling in application development

## Validation & Encoding for Android

Input validation and output encoding can help ensure that data and networks are kept secure. By understanding the various methods of exploit, mobile developers can help prevent such attacks. This course will help attendees understand how to validate and encode information on the Android platform. This self-paced, e-Learning course provides an overview of best practices for input validation and output encoding on the Android platform.

**Course Duration:** 30 minutes

**Intended Audience:** Mobile Application Developers, Software Developers, Security Professionals, Penetration Testers

### Lesson 1: Protection against Injection

**Objectives:** After completing this lesson, you should be able to:

- Understand how lack of input validation can be exploited
- Explain how to encode untrusted data for display

### Lesson 2: Validating Data in Interprocess Communications

**Objectives:** After completing this lesson, you should be able to:

- Describe methods Android provides for interprocess communications (IPC)
- Understand the best practices for securing IPC's

### Lesson 3: Validating Data from Third-Party Web Services

**Objectives:** After completing this lesson, you should be able to:

- Explain how enterprise web services can be exploited
- Describe the impact of not using customer permissions

## AppSec Tutorials

AppSec Tutorials provide succinct, in 15 minutes or less, on-demand Security Education trainings for software developers. These trainings can help developers understand how common application security vulnerabilities are exploited and how to protect against them.

### AppSec Tutorial – CRLF Injection

**Course Duration:** 15 minutes

**Course Summary:** A CRLF injection refers to the special character elements "Carriage Return" and "Line Feed" and is a vulnerability that occurs when an attacker injects a CRLF character sequence where it is not expected. CRLF injection exploits security vulnerabilities at the application layer. By exploiting the CRLF injection flaw in an HTTP response for example, attackers can modify application data, compromising integrity and enable further exploitation through other vulnerabilities.

### AppSec Tutorial – Cross Site Scripting (XSS)

**Course Duration:** 10 minutes

**Course Summary:** Cross Site Scripting (XSS) is one of the most common vulnerabilities in web applications. An XSS vulnerability arises when web applications take data from users and dynamically include it in web pages without first properly validating the data. XSS vulnerabilities allow an attacker to execute arbitrary commands and display arbitrary content in a victim user's browser. A successful XSS attack leads to an attacker controlling the victim's browser or account on the vulnerable web application. This AppSec Tutorial demonstrates a Cross Site Scripting attack on a Java web application and a .NET web application; and provides a demonstration of remediation for a Java web application and a .NET web application.

### AppSec Tutorial – Directory Traversal

**Course Duration:** 10 minutes

**Course Summary:** Directory Traversal (also known as "path traversal") is one of the most prevalent software weaknesses, and if exploited correctly, could potentially allow an attacker to view restricted files or other sensitive information. This AppSec Tutorial explains how to exploit a Directory Traversal flaw within a web application, and provides guidance for remediating this flaw in both Java and .NET.

### AppSec Tutorial – Operating System Command Injection

**Course Duration:** 15 minutes

**Course Summary:** "Operating System Command Injection" is a type of application security vulnerability in the family of Injection flaws. The presence of this flaw in an application's code could allow malicious users to cause far-reaching damage. In this course, you'll see an example of how this weakness can be exploited, and then how it can be fixed.

## AppSec Tutorial – SQL Injection

Course Duration: 10 minutes

Course Summary: SQL Injection is the most widely seen and exploited injection vulnerability. This AppSec Tutorial demonstrates an SQL Injection attack on a Java web application and a .NET web application; and provides a demonstration of remediation for a Java web application and a .NET web application. Injection is a very damaging vulnerability, number 1 in the both the OWASP Top Ten 2010 and the OWASP Top Ten 2013. SQL injection is a vulnerability in which an attacker is able to submit a database SQL command that is executed by a web application, exposing the back-end database. SQL injection attacks can occur when a web application utilizes user-supplied data without proper validation or encoding as part of a command or query.

Local Region representative please contact E-SPIN Group veracode@e-spincorp.com


E-SPIN Group of Companies (Business Centre strategic presence across the region)
E-SPIN SDN. BHD. (714753-U) (GST No: 001328111616)
E-SPIN INTERNATIONAL LIMITED 億轉國際有限公司 (1970945)
E-SPIN INTERNATIONAL PTE. LTD. (201312412W)

Malaysia (cover Malaysia & Brunei)
No. 21-2, Jalan PJU 8/3B,
Perdana Business Centre,
Damansara Perdana
47820 Petaling Jaya, Selangor
Malaysia
Tel: +603 9212 7768, +603 7725 4767
Fax: +603 7725 4757

Hong Kong (cover Hong Kong, Macau, Taiwan, Japan, Korea and International Trade)
Hong Kong Island
Room 1104, Crawford House,
70 Queen Road Central, Central,
Hong Kong
Tel: +852 5801 4153
Fax: +852 3010 7118

Singapore (cover Singapore and International Trade)
10 Anson Road
#26-10 International Plaza
Singapore 079903
Tel: +65 3158 2203
Fax: +65 6338 6311

China (cover Greater China Region, Mongolia)
15/F L`Avenue,
99 Xianxia Road, Chang Ning District,

China (cover Greater China Region, Mongolia)
15/F L`Avenue,
99 Xianxia Road, Chang Ning District,
Shanghai 200051
China
Tel: +861087833348

Indonesia
Office 8, Level 18-A, Jalan Jend Sudirman Kav. 52-53
Sudirman Central Business District (SCBD)
Jakarta Selatan
Daerah Khusus Ibukota Jakarta 12190
Indonesia
Tel: +6221 3002 4443

Philippines (cover Philippines, Guam & Palau)
Penthouse Level, Mavenue Building,
7844 Makati Avenue,
Makati City, Metro Manila,
1209 Philippines
Tel: +6322312189

Thailand (cover Thailand, Myanmar (Burma) / Indochina - Vietnam, Cambodia & Laos)
195 Unit 4703, 47th Floor, Empire Tower,
South Sathorn Road, Yannawa, Sathorn,
Bangkok 10120 Thailand
Tel: +66 60 002 4168