SECURE CODE WARRIOR

Secure Code Warrior

Learning Content

knowbe4

Introduction

Secure Code Warrior content available through KnowBe4 provides foundational secure coding and AI governance awareness for organizations with technical roles. Available to KnowBe4 Diamond customers, it covers OWASP Top 10 risks, AI-assisted development, and modern secure coding practices. The content includes **31 learning activities** across 9 structured learning series—combining videos, guidelines, interactive walkthroughs and missions to reinforce both understanding and practical skill development. Interactive walkthroughs and missions are available across **10 programming languages**, and all content is delivered in **8 spoken languages** to support global teams. **This guide outlines the content included in each series and maps it to relevant roles to help teams structure learning and build capability over time.**

Table of Contents

Summary: Learning Content by Role

OWASP Web Risk Coverage

Engineers, Architects, QA, Product Managers

OWASP A01: Broken Access Control

- Video: Missing Function Level Access Control
- Walkthrough: Missing Function Level Access Control
- Video: Path Traversal
- Walkthrough: Path Traversal

OWASP A02: Security Misconfiguration

- Video: XML External Entities (XXE)
- Walkthrough: XML External Entities (XXE)
- Video: Disabled Security Features
- Walkthrough: Disabled Security Features

OWASP A03: Software Supply Chain Failures

- Guideline: Introduction to Open Source Software
- Walkthrough: Using Known Vulnerable Components
- Guideline: Overview of OWASP Top 10 Risk for Open Source Software
- Guideline: Managing Open-Source Software Risks

OWASP A04: Cryptographic Failures

- Video: Plain text Storage of Passwords
- Walkthrough: Plain text Storage of Passwords
- Video: Weak Algorithm Use
- Walkthrough: Weak Algorithm Use

OWASP AI Risk Coverage

Engineers, Architects, QA, Product Managers

OWASP Top 10 for Large Language Model Applications

- Video: Direct Prompt Injection
- Video: Sensitive Information Disclosure
- Video: Supply Chain

AI-Assisted Development & AI Security Roles

Engineers, Architects, Product Managers, AI tool users

Coding with AI

- Video: Potential Dangers of Using AI/LLMs when writing code
- Video: Advantages of Using AI/LLMs when writing code

OWASP Top 10 for Large Language Model Applications

- Video: Direct Prompt Injection
- Video: Sensitive Information Disclosure
- Video: Supply Chain

Backend Engineering Teams

Backend Engineers, API Engineers

OWASP A01: Broken Access Control

- Video: Missing Function Level Access Control
- Walkthrough: Missing Function Level Access Control
- Video: Path Traversal
- Walkthrough: Path Traversal

OWASP A02: Security Misconfiguration

- Video: XML External Entities (XXE)
- Walkthrough: XML External Entities (XXE)
- Video: Disabled Security Features
- Walkthrough: Disabled Security Features

OWASP A03: Software Supply Chain Failures

- Guideline: Introduction to Open Source Software
- Walkthrough: Using Known Vulnerable Components
- Guideline: Overview of OWASP Top 10 Risk for Open Source Software
- Guideline: Managing Open-Source Software Risks

OWASP A04: Cryptographic Failures

- Video: Plain text Storage of Passwords
- Walkthrough: Plain text Storage of Passwords
- Video: Weak Algorithm Use
- Walkthrough: Weak Algorithm Use

Data Protection

- Video: Data Protection
- Video: Sensitive Data Exposure
- Video: Plain Text Storage of Sensitive Information
- Mission: Sensitive Data Exposure

Web Application Security 101

- Video: Issues with Client Side Security Measures
- Video: Local Storage
- Video: In-depth into Cookies and Sessions
- Walkthrough: Exposed Session Token

Frontend Web Engineering Teams

Frontend Engineers

OWASP A03: Software Supply Chain Failures

- Guideline: Introduction to Open Source Software
- Walkthrough: Using Known Vulnerable Components
- Guideline: Overview of OWASP Top 10 Risk for Open Source Software
- Guideline: Managing Open-Source Software Risks

Web Application Security 101

- Video: Issues with Client Side Security Measures
- Video: Local Storage
- Video: In-depth into Cookies and Sessions
- Walkthrough: Exposed Session Token

Software Architecture & Technical Leadership

Architects, Engineering Managers, Lead Engineers

OWASP A01: Broken Access Control

- Video: Missing Function Level Access Control
- Walkthrough: Missing Function Level Access Control
- Video: Path Traversal
- Walkthrough: Path Traversal

OWASP A02: Security Misconfiguration

- Video: XML External Entities (XXE)
- Walkthrough: XML External Entities (XXE)
- Video: Disabled Security Features
- Walkthrough: Disabled Security Features

OWASP A03: Software Supply Chain Failures

- Guideline: Introduction to Open Source Software
- Walkthrough: Using Known Vulnerable Components
- Guideline: Overview of OWASP Top 10 Risk for Open Source Software
- Guideline: Managing Open-Source Software Risks

OWASP A04: Cryptographic Failures

- Video: Plain text Storage of Passwords
- Walkthrough: Plain text Storage of Passwords
- Video: Weak Algorithm Use
- Walkthrough: Weak Algorithm Use

Data Protection

- Video: Data Protection
- Video: Sensitive Data Exposure
- Video: Plain Text Storage of Sensitive Information
- Mission: Sensitive Data Exposure

Web Application Security 101

- Video: Issues with Client Side Security Measures
- Video: Local Storage
- Video: In-depth into Cookies and Sessions
- Walkthrough: Exposed Session Token

Threat Modeling

- Video: Threat Modeling Overview
- Video: S.T.R.I.D.E

OWASP Top 10 for Large Language Model Applications

- Video: Direct Prompt Injection
- Video: Sensitive Information Disclosure
- Video: Supply Chain

Coding with AI

- Video: Potential Dangers of Using AI/LLMs when writing code
- Video: Advantages of Using AI/LLMs when writing code

Data & Analytics Teams

Data Scientists, Data Analysts, Data Engineers

Data Protection

- Video: Data Protection
- Video: Sensitive Data Exposure
- Video: Plain Text Storage of Sensitive Information
- Mission: Sensitive Data Exposure

OWASP A04: Cryptographic Failures

- Video: Plain text Storage of Passwords
- Walkthrough: Plain text Storage of Passwords
- Video: Weak Algorithm Use
- Walkthrough: Weak Algorithm Use

OWASP A03: Software Supply Chain Failures

- Guideline: Introduction to Open Source Software
- Walkthrough: Using Known Vulnerable Components
- Guideline: Overview of OWASP Top 10 Risk for Open Source Software
- Guideline: Managing Open-Source Software Risks

Quality Engineering & Testing Teams

QA Engineers, QA Managers, Testers

OWASP A01: Broken Access Control

- Video: Missing Function Level Access Control
- Walkthrough: Missing Function Level Access Control
- Video: Path Traversal
- Walkthrough: Path Traversal

Data Protection

- Video: Data Protection
- Video: Sensitive Data Exposure
- Video: Plain Text Storage of Sensitive Information
- Mission: Sensitive Data Exposure

Web Application Security 101

- Video: Issues with Client Side Security Measures
- Video: Local Storage
- Video: In-depth into Cookies and Sessions
- Walkthrough: Exposed Session Token

OWASP Top 10 for Large Language Model Applications

- Video: Direct Prompt Injection
- Video: Sensitive Information Disclosure
- Video: Supply Chain

Product & Technical Business Roles

Product Managers, Project Managers, Business Analysts

Threat Modeling

- Video: Threat Modeling Overview
- Video: S.T.R.I.D.E

OWASP Top 10 for Large Language Model Applications

- Video: Direct Prompt Injection
- Video: Sensitive Information Disclosure
- Video: Supply Chain

Summary: Learning Content by Series

OWASP 2025 A01 - Broken Access Control

Available in: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series explores the critical risks associated with Broken Access Control by examining how improper authorization checks and insufficient input validation allow users to exceed their intended permissions. Through video and hands-on walkthrough activities, developers will learn about two common Broken Access Control vulnerabilities. They will learn how attackers gain access to restricted data and server files, and how to mitigate these vulnerabilities with server-side role-based access control and input sanitization.

Video: Missing Function Level Access Control

4 min Video Module

This lesson defines missing function-level access control and explores how improper authorization checks allow users to access restricted data or administrative functions. Learners will examine real-world examples such as forced browsing and unauthorized request crafting to understand the potential for account takeovers and data theft. The module provides an overview of essential prevention strategies, emphasizing server-side role-based access control and the principle of denying access by default.a

Walkthrough: Missing Function Level Access Control

5 min Training Module

This hands-on sandbox lesson focuses on identifying and exploiting missing function-level access control within a web application. Learners will investigate how hiding sensitive UI elements in code comments fails to provide actual security against unauthorized access to administrative endpoints. The activity guides participants through using browser tools to discover hidden features and demonstrates how to bypass insufficient authorization checks to gain access to restricted user data.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Video: Path Traversal

3 min Video Module

This lesson provides an overview of path traversal vulnerabilities, also known as directory traversal or dot-dot-slash attacks. Learners will explore how attackers manipulate URL path inputs to access sensitive files outside of the web root folder through techniques like directory climbing. The video also provides an overview of prevention strategies, such as normalizing user input, using chroot jails, and implementing strict code access policies to restrict unintended file system access.

Walkthrough: Path Traversal

5 min Training Module

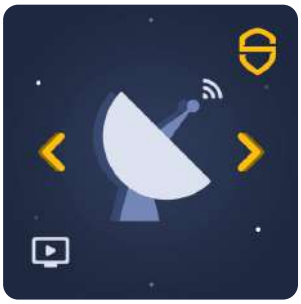
This hands-on sandbox lesson focuses on identifying and exploiting path traversal vulnerabilities within a web application's file download functionality. Learners will investigate how a lack of input validation on file name parameters can allow an attacker to use dot-dot-slash sequences to break out of intended directories and access sensitive server files like database backups. The activity emphasizes the critical importance of sanitizing user-controlled input and implementing strict validation to prevent unauthorized file access and potential malicious code execution.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

OWASP 2025 A02 - Security Misconfiguration

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

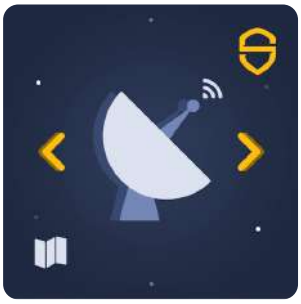
This series explores the risks of Security Misconfiguration by examining how improperly hardened application settings and weakly configured parsers create critical vulnerabilities. Through video and hands-on walkthrough activities, developers will learn about two common Security Misconfiguration vulnerabilities. They will learn how attackers gain access to sensitive files and credentials through misconfiguration, and learn how to mitigate these risks by disabling unnecessary features and implement secure configuration standards.



Video: XML External Entities (XXE)

3 min Video Module

This lesson provides an introduction to XML External Entity (XXE) injection, explaining how attackers exploit weakly configured XML parsers to access sensitive server files or perform internal port scanning. Learners will examine the causes of XXE vulnerabilities, such as improperly sanitized input, and explore the potential impact on data security and system availability. The video also provides essential prevention strategies, including the use of allowlist input validation and the configuration of XML parsers to disable external entity support.



Walkthrough: XML External Entities (XXE)

5 min Training Module

This hands-on sandbox lesson focuses on identifying and exploiting XML External Entity (XXE) vulnerabilities within a banking application. Learners will explore how improperly configured XML processors can be tricked into disclosing sensitive system files, such as /etc/passwd, by manipulating DTD processing and external entity declarations. The activity emphasizes the importance of disabling DTD definition processing and using safe built-in library functions to protect against unauthorized file access and server-side request forgery.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Video: Disabled Security Features

2 min Video Module

This lesson provides an overview of the risks associated with disabled security features, exploring how deactivating protective settings can leave applications vulnerable to exploitation. Learners will examine a real-world example involving insecure cookie flags and see how these oversights allow attackers to capture session data and perform unauthorized actions. The video also covers best practices for verifying active security restrictions and implementing compensatory checks when features must be disabled for operational reasons.

Walkthrough: Disabled Security Features

5 min Training Module

This hands-on sandbox lesson explores the risks of disabled security features, specifically focusing on how enabling static file browsing can lead to unauthorized directory traversal. Learners will perform reconnaissance to locate sensitive backup files and extract default administrator credentials that were left exposed in plain text. The activity demonstrates how to disable directory listings or utilize default index files to prevent attackers from browsing server contents and compromising the system.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

OWASP 2025 A03 - Software Supply Chain Failures

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series explores supply chain security challenges of modern development by examining how vulnerabilities and malicious changes in third-party dependencies can compromise software. Through detailed guidelines and a hands-on walkthrough activity, developers will learn to identify risks associated with open-source software, including unmaintained components and common known exploits such as remote code execution. They will learn how to mitigate these threats by implementing software composition analysis, maintaining a software bill of materials, and establishing robust component policies to secure their entire development life cycle.

Guideline: Introduction to Open Source Software

5 min Training Module

This lesson provides an introduction to Open Source Software (OSS), exploring its collaborative development model and the critical role it plays in modern application security. Learners will examine the review and maintenance processes used by communities to ensure software quality, alongside an overview of common permissive and copyleft licensing structures. The course also highlights inherent security challenges, such as vulnerability management and licensing compliance, to help organizations effectively mitigate risks when integrating open source components into the software development life cycle.

Walkthrough: Using Known Vulnerable Components

5 min Training Module

This hands-on sandbox lesson explores the risks of using vulnerable versions of open source software by examining the Apache path traversal and remote code execution vulnerabilities tracked as CVE-2021-41773 and CVE-2021-42013. The activity demonstrates the critical importance of software supply chain management, demonstrating how regular patching schedules and secure configuration hardening are essential to protecting infrastructure from publicly documented vulnerabilities.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Guideline: Overview of OWASP Top 10 Risk for Open Source Software

5 min Training Module

This lesson provides an overview of the OWASP Top 10 risks associated with the use of open source software (OSS), moving beyond simple vulnerability management to address legal, operational, and supply chain threats. Learners will examine the specific impacts of risks such as known vulnerabilities, unmaintained software, and name confusion attacks while analyzing real-world examples like Heartbleed and the XZ Utils incident. The module outlines effective mitigation strategies, including the use of Software Bill of Materials (SBOM) and Software Composition Analysis (SCA) tools, to help development teams secure their software supply chain.

Guideline: Managing Open-Source Software Risks

5 min Training Module

This lesson explores the strategies and technologies required to effectively manage risks associated with Open-Source Software (OSS) within a Cyber Supply Chain Risk Management framework. Learners will examine how to develop a comprehensive OSS component policy that covers selection criteria, contribution guidelines, and internal project governance. The course further details the operational role of Software Composition Analysis and Software Bill of Materials in automating vulnerability detection and ensuring license compliance across the software supply chain.

OWASP 2025 A04 - Cryptographic Failures

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series explores the critical risks of Cryptographic Failures by examining how plain text storage and weak hashing algorithms leave sensitive data vulnerable to exposure. Through videos and hands-on walkthrough activities, developers will learn how attackers exploit improperly secured credentials and use rainbow tables to compromise database contents. They will learn how to mitigate these vulnerabilities by implementing strong standard algorithms, salts, and resource-heavy hashing libraries to ensure the integrity and confidentiality of user data.

Video: Plain text Storage of Passwords

3 min Video Module

This lesson provides an overview of sensitive data exposure, focusing on the risks of storing credentials without adequate protection. Learners will examine how weak hashing and a lack of encryption can lead to identity hijacking or financial loss. The module provides an overview of practical prevention strategies, including the use of strong hashing algorithms, salts, and secure transport protocols like TLS to safeguard sensitive information.

Walkthrough: Plain text Storage of Passwords

5 min Training Module

This hands-on sandbox lesson focuses on the risks associated with storing sensitive data in plain text, guiding learners through the process of identifying and exploiting improperly secured credentials. Learners will explore a vulnerable web application to discover how concealed password fields can be revealed through browser developer tools and the inspection of HTML source code. Secure code training is integrated by highlighting the necessity of strong hashing algorithms, salting, and peppering to ensure that even administrators cannot access confidential user passwords.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Video: Weak Algorithm Use

2 min Video Module

This lesson explores the risks associated with using weak cryptographic algorithms for storing sensitive data, such as personal information and passwords. Learners will examine how malicious insiders or external attackers can exploit insecure storage to gain unauthorized access to database contents. The video also provides an overview of essential prevention steps, including the use of strong standard algorithms, proper salting for password hashes, and effective key management.

Walkthrough: Weak Algorithm Use

5 min Training Module

This hands-on walkthrough focuses on the risks associated with weak hashing algorithms and the use of rainbow tables to compromise sensitive data. Learners will investigate a simulated database leak to identify the administrator's password by comparing SHA256 hashes against a list of precomputed plaintext values. The lesson emphasizes the importance of robust security measures, such as using salted passwords and intentionally resource-heavy hashing libraries, to defend against common credential attacks.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Data Protection

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series explores the fundamental principles of data protection by examining how the improper handling of personal, payment, and regulated information leads to unauthorized exposure. Through videos and a hands-on walkthrough activity, developers will learn how to identify data leakage in API responses and the risks associated with storing sensitive details in plain text. They will learn how to mitigate these vulnerabilities by implementing encryption, secure communication channels, and strict data filtering to ensure only necessary information is transmitted and stored.

Video: Data Protection

5 min Video Module

This lesson explores the fundamental principles of data protection by analyzing common vulnerabilities and their real-world consequences. Learners will examine how to secure sensitive information through encryption, secure communication channels like TLS, and source code obfuscation. The video also highlights best practices for maintaining user privacy and implementing strict access controls to prevent unauthorized data exposure.

Video: Sensitive Data Exposure

2 min Video Module

This lesson provides an overview of sensitive data exposure, defining how vulnerabilities occur when personal, payment, or regulated data is improperly handled. Learners will explore a real-world example of data leakage through server responses and the potential risks of exposing information to unauthorized parties. The module also provides an overview of essential prevention strategies, including data identification, minimizing data transmission, and the use of encryption to ensure compliance and security.

Video: Plain Text Storage of Sensitive Information

2 min Video Module

This lesson provides an overview of the risks associated with storing sensitive data in clear text format, which can lead to identity theft and unauthorized data access. Learners will examine a real-world scenario involving a database breach to understand the consequences of failing to encrypt personal information. The video also highlights the importance of using strong, community-accepted encryption algorithms to secure sensitive details prior to storage.

Mission: Sensitive Data Exposure

5 min Video Module

This hands-on sandbox lesson explores the risks of sensitive data exposure through a faulty API implementation that inadvertently leaks user credentials. Learners will step into the role of an attacker to intercept authentication tokens and exploit a vulnerable endpoint that exposes an employee's plain-text password and personal details. The activity emphasizes secure coding principles by demonstrating the importance of data filtering, returning only necessary information in API responses, and the critical need for hashing passwords before storage.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Web Application Security 101

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series introduces the fundamentals of web application security by examining the risks associated with client-side storage, session management, and the limitations of front-end security measures. Through videos and a hands-on walkthrough activity, developers will learn how attackers bypass local restrictions and exploit exposed session tokens to perform account hijacking. They will learn how to mitigate these foundational vulnerabilities by implementing server-side validation, securing cookie configurations, and avoiding the transmission of sensitive identifiers in URLs.

Video: Issues with Client Side Security Measures

3 min Video Module

This lesson explores the risks associated with relying exclusively on client-side security mechanisms like input validation and hard-coded passwords. Learners will examine how attackers use local proxies and source code inspection to bypass these restrictions, potentially leading to injection attacks or unauthorized access. The module also demonstrates why security-related calculations must be performed on the server side to ensure robust application protection.

Video: Local Storage

3 min Video Module

This lesson provides an overview of local storage vulnerabilities, explaining how web applications use client-side key-value pairs for persistence and session management. Learners will examine how sensitive data stored in the browser can be compromised through physical access or cross-site scripting (XSS) attacks. The module also provides an overview of essential prevention strategies, such as avoiding the storage of sensitive information and implementing robust input sanitization.

Video: In-depth into Cookies and Sessions

5 min Video Module

This lesson provides an overview of how cookies and sessions are used to maintain state in the naturally stateless HTTP protocol. Learners will identify common vulnerabilities caused by missing or misconfigured cookie flags, such as Secure and HttpOnly, and understand how session identifiers can be compromised if passed as URL parameters. The module also provides an overview of best practices for developers to properly configure cookie headers and limit session exposure to prevent unauthorized account access.

Walkthrough: Exposed Session Token

5 min Training Module

This hands-on sandbox lesson explores the risks of exposing sensitive information in URLs, specifically focusing on how session tokens can be leaked through query parameters. Learners will perform a session hijacking attack by inducing a user to share a receipt link, capturing their session ID, and using a cookie editor to take over the unauthorized account. The activity demonstrates why using session identifiers for authorization in web addresses is a dangerous practice that increases the likelihood of account compromise and data exposure.

Available in 10 programming languages—C# (.NET Basic, .NET Core), Go, Java (Enterprise Edition, Spring), JavaScript (Node.js / Express), Python (Basic, Django), PHP, and pseudocode (web-based scenarios)

Threat Modeling

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series introduces threat modeling as a structured approach for identifying and understanding security risks throughout the development process. Developers will learn the core elements of a systematic assessment and explore the STRIDE methodology to categorize threats against critical system properties like confidentiality and integrity. They will learn how to apply these concepts to analyze practical examples, such as repudiation threats, ensuring a proactive defense against potential application vulnerabilities.

Video: Threat Modeling Overview

3 min Video Module

This lesson provides an overview of threat modeling as a structured way to identify and understand security risks. Learners will define the four core elements common to the most-used threat modeling methodologies and explore the basic steps of a systematic assessment process.

Video: S.T.R.I.D.E

4 min Video Module

This lesson introduces the STRIDE threat modeling methodology. Learners will explore the six threat categories and learn how they relate to critical system properties like confidentiality and integrity. The module demonstrates how to apply these concepts through practical examples, such as analyzing repudiation threats within a payment application.

OWASP Top 10 for Large Language Model Applications

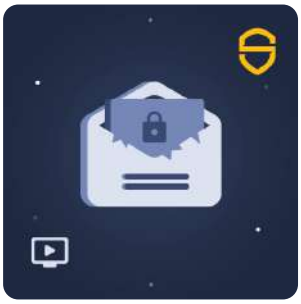
Available languages: English (United States)

This series explores the unique security risks associated with Large Language Model applications by examining how malicious inputs and supply chain weaknesses can compromise AI systems. Developers will learn about three significant vulnerabilities: direct prompt injection, sensitive information disclosure, and supply chain tampering. They will learn how to mitigate these threats by implementing input screening, data filtering, and AI Bills of Materials to ensure the integrity and privacy of their AI-driven applications.

Video: Direct Prompt Injection

4 min Video Module

This lesson provides an overview of prompt injection vulnerabilities within AI systems, explaining how malicious inputs can manipulate or override large language model instructions. Learners will explore the differences between direct and indirect injection attacks, including real-world scenarios like link traps and account hijacking. The module also provides an overview of defense strategies such as input screening, the principle of least privilege, and the importance of separating user input from system directives.



Video: Sensitive Information Disclosure

3 min Video Module

This lesson provides a concise overview of how sensitive information can be unintentionally exposed by AI systems and large language models. Learners will identify the risks associated with the entry and disclosure phases of data handling, including training data vulnerabilities and malicious user prompts. The module also provides an overview of practical mitigation strategies such as data filtering, the rule of least privilege, and output monitoring to ensure compliance with privacy standards.

Video: Supply Chain

4 min Video Module

This lesson provides an overview of supply chain vulnerabilities within the context of software development and large language models. Learners will explore the key stages of the AI supply chain, including datasets, model serialization, and training frameworks, while identifying risks such as poisoned data and unauthorized tampering. The module also provides an overview of essential defense strategies, such as using AI Bills of Materials and cryptographic signatures to ensure the integrity of digital assets.

Coding with AI

Available languages: English (United States), Chinese (Mandarin) - Simplified, Chinese (Mandarin) - Traditional, French (Europe), German, Japanese, Korean, Spanish (Europe)

This series explores the balance between the efficiency gains and security risks of integrating artificial intelligence into the software development lifecycle. Developers will learn about the advantages of AI-assisted code generation and pair programming alongside critical dangers such as code hallucinations, proprietary data exposure, and the introduction of insecure dependencies. They will learn how to responsibly leverage AI tools as a collaborative resource while maintaining the human oversight necessary to validate code quality and ensure long-term security.

Video: Potential Dangers of Using AI/LLMs when writing code

16 min Video Module

This lesson explores the critical security and ethical risks associated with using artificial intelligence for code generation, such as the accidental exposure of sensitive proprietary data and the integration of insecure or outdated code. Learners will examine how AI hallucinations and a lack of contextual understanding can introduce unique vulnerabilities, including malicious library references and logic flaws. The module also addresses long-term concerns like the reinforcement of training biases and the potential decline in developer problem-solving skills due to over-reliance on automated tools.

Video: Advantages of Using AI/LLMs when writing code

9 min Video Module

This lesson explores the advantages of using artificial intelligence and large language models to assist in the software development lifecycle. Learners will discover how AI tools can accelerate code generation, serve as a collaborative pair programmer for debugging, and provide a personalized learning experience for complex technical topics. The module emphasizes the importance of balancing these efficiency gains with developer responsibility, highlighting the need for human review to validate code quality and ensure security.