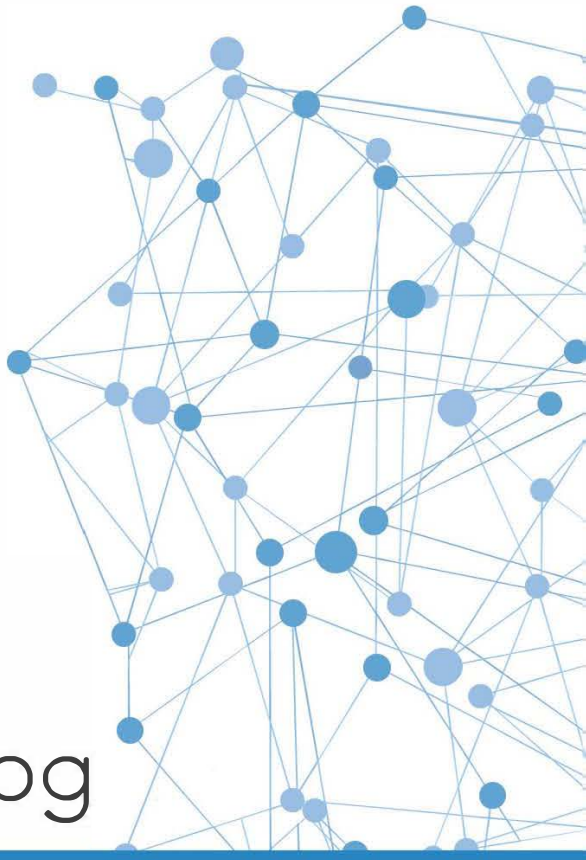
SECURITY INNOVATION



Training Program Catalog

All Courses

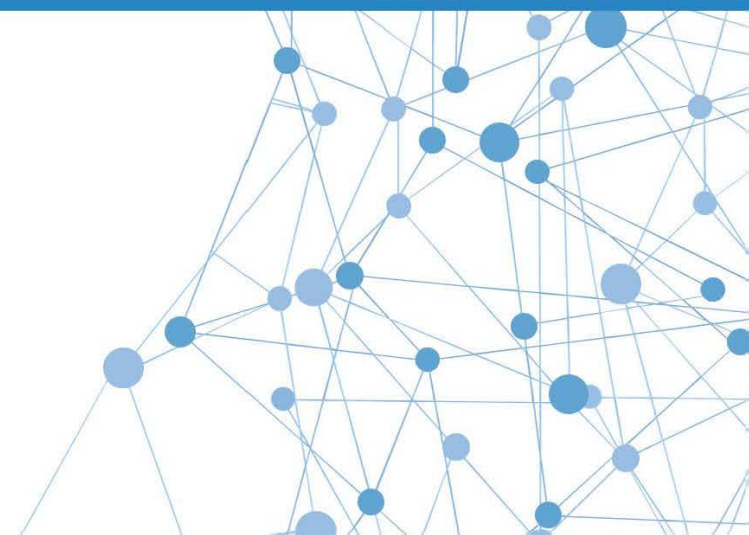


Table of Contents

Computer Based Training - Security Awareness - General Staff

Computer Based Training - Secure Coding

Computer Based Training - Secure Design

Computer Based Training - Secure Engineering

Computer Based Training - Secure Testing

Security Essentials - NEW

Instructor Led Training

AWA 007
Information Privacy and Security Awareness for Executives
Duration: 45 minutes

This course provides decision-makers and managers with a concise summary essential ISPA requirements. Content is aligned with the topics contained in our standard ISPA courses, ensuring managers and staff are focused on the same objectives.

AWA 008
Information Privacy - Classifying Data
Duration: 15 minutes

This introductory course is designed for general staff in roles such as human resources, legal, marketing, finance, sales, operations and customer service. This course equips employees to recognize the importance of understanding what constitutes private data.

AWA 009
Information Privacy - Protecting Data
Duration: 20 minutes

This introductory course is designed for general staff in roles such as human resources, legal, marketing, finance, sales, operations and customer service. This course equips employees to recognize the importance of understanding what constitutes private data and how to behave in a proactive manner to protect this information in their everyday work

---

AWA 010
Email Security
Duration: 10 minutes

This security awareness course is intended to teach students how to recognize malicious email before it can become a threat, how to properly handle email, and best practices around how and when to use email to send specific types of information. Through participating in this courses, students will be able to define Personally Identifiable Information (PII), understand the impact of sending sensitive information over an insecure medium, and identify information that should not be sent by email.

AWA 012
Malware Awareness
Duration: 10 minutes

This security awareness course is intended to teach students how to identify and define types of malware. Through participating in this course, students will be able to recognize evidence of active infection and understand what the proper actions are to prevent such attacks.

AWA 013
Mobile Security
Duration: 15 minutes

This security awareness course is intended to give students a look at mobile device security. Through participating in this course, students will be able to list the characteristics of mobile device platforms and identify the role device ownership plays as a basis for understanding application risk.

AWA 014
Password Security
Duration: 10 minutes

This security awareness course is intended to teach students how to create and remember strong passwords, therefore eliminating the need to use insecure practices. Through participating in this course, students will learn how to recognize the risks surrounding password security, identify safeguards used to protect passwords, and summarize techniques used by attackers to obtain passwords.

AWA 015
PCI Compliance
Duration: 15 minutes

This security awareness course is intended to teach students to follow the PCI Security Standards in order to understand how to identify different types of sensitive data and handle it properly. Through participating in this course, students will be able to recognize appropriate protection mechanisms for cardholder data and acknowledge how to the PCI DSS helps minimize risk to cardholder data.

AWA 016
Phishing Awareness
Duration: 10 minutes

This security awareness course is intended to teach students how to recognize malicious email before it can become a threat. Through participating in this course, students will be able to understand the various ways in which attackers try to trick and entice users to trigger malicious events through email, as well as best practices to properly handle and avoid phishing attacks.

AWA 017
Physical Security
Duration: 10 minutes

This course is intended to teach students accepted practices for minimizing breaches and give them the ability to identify different types of data that may be exposed via hardware theft. Through participating in this course, students will be able to understand what physical security is and why it is everyone's responsibility, identify common physical security attacks, and identify physical security best practices.

AWA 018
Social Engineering Awareness
Duration: 15 minutes

This security awareness course is intended to teach students how to identify the many forms of social engineering and its potential impacts. Through participating in this course, students will be able to identify techniques used by social engineers and understand how to establish validity of requests in order to perform daily business functions in light of potential threats.

AWA 019
Travel Security
Duration: 15 minutes

This security awareness course is intended to introduce students to the risks associated with transporting sensitive data. Through participating in this course, students will be able to recognize threats that may be present while traveling, identify the risks certain locations may harbor, and understand the defenses that you may employ while traveling.

AWA 101
Fundamentals of Application Security
Duration: 60 minutes

This course introduces the fundamentals of application security. It discusses the main drivers for application security, fundamental concepts of application security risk management, the anatomy of an application attack, some common attacks, the concept of input validation as a primary risk mitigation technique, and key security principles and best practices for developing secure applications.

COD 101
Fundamentals of Secure Development
Duration: 60 minutes

This course introduces you to the need for secure software development, as well as the models, standards, and guidelines that you can use to understand security issues and improve the security posture of your applications. It also describes key application security principles and secure coding principles, and explains how to integrate secure development practices into the software development lifecycle. Note: The training should take approximately 80 minutes to complete. This course is optimally viewed at a screen resolution of 1024x768.

COD 110
Fundamentals of Secure Mobile Development
Duration: 60 minutes

This course introduces developers to the common risks associated with Mobile applications including client side injection, sensitive data handling, network transition, application patching, web based attacks, phishing, third-party code, location security and privacy and denial of service. The student is then given an overview of the Mobile application development best practices to reduce these risks including input validation, output encoding, least privilege, code signing, data protection at rest and in transit, avoiding client side validation, and using platform security capabilities as they apply in mobile environments. Included is a discussion of threat modeling mobile applications. With knowledge checks throughout, the student who completes this course will have an understanding of mobile environment threats and risks, and the programming principles to use to address them.

COD 141
Fundamentals of Secure Database Development
Duration: 110 minutes

In practice, the database represents the goal of many attackers, as this is where the information of value is maintained. However, functional requirements and security testing often focus on the interaction between a software user and the application, while the handling of data is assumed to be secure. This course is platform and technology agnostic, and will provide software architects and developers with an understanding of database development best practices.

COD 152
Fundamentals of Secure Cloud Development
Duration: 30 minutes

This course introduces developers to the common risks associated with Cloud applications, including the security features of the different series models (IaaS, PaaS, and SaaS), how to identify and mitigate the most common vulnerabilities, the unique security challenges of "Big Data", and how to apply the Microsoft SDL to cloud applications. Threat coverage includes unauthorized account access, insecure APIs, shared technology, data leakage, and account hijacking, as well the importance of complying with regulatory requirements.

COD 153
Fundamentals of Secure AJAX Code
Duration: 35 minutes

This course introduces security issues and challenges specific to AJAX applications. It provides an overview of AJAX technology, and presents common AJAX application vulnerabilities and attack vectors. Upon completion of this class, participants will be able to identify the differences between regular and AJAX applications, common AJAX vulnerabilities that attackers tend to exploit, and major threats to AJAX applications from cross-site scripting, cross-site request forgery, and injection attacks.

COD 160
Fundamentals of Secure Embedded Software Development
Duration: 90 minutes

In this course, you will learn about security issues inherent to embedded device architecture. You will also learn about techniques to identify system security and performance requirements, develop appropriate security architecture, select the correct mitigations, and develop policies that can ensure the secure operation of your system.

COD 170
Identifying Threats to Mainframe COBOL Applications and Data
Duration: 20 minutes

This course covers the most common security issues that affect the confidentiality, integrity and availability of COBOL programs on mainframes. These include SQL Injection, Command Injection, Integer Overflow, Weak Cryptography, Unencrypted Communications and Race Conditions.

COD 190
IoT Embedded Systems Security - Fundamentals of Secure Mobile Development
Duration: 30 minutes

This course module provides additional training on Secure Mobile Development of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

COD 200
Creating Secure C Code SERIES NEW
Duration: 30 minutes

This series provides C developers with the knowledge and skills required to secure communications with Transport Layer Security (TLS) and to implement run-time protections with technologies such as stack security cookies, Address Space Layout Randomization (ASLR), and No-eXecute.

COD 201
Secure C Encrypted Network Communications NEW
Duration: 15 minutes

In this course, you will learn about secure communications using Transport Layer Security (TLS), and best practices for implementing these with your C and C++ applications. After completing this course, you will be able to identify the basic principles of TLS, identify libraries and interfaces for implementing the TLS protocol, identify TLS security considerations, and identify alternatives to TLS.

COD 202
Secure C Run-Time Protection NEW
Duration: 15 minutes

This course discusses common run-time protection technologies that you can use to protect your application from attacks. After completing this course , you will be able to identify run-time protection technologies, such as stack security cookies, address space layout randomization, and No-eXecute. You will also be able to identify their limitations, and how to apply them to your applications.

COD 205

Creating Secure C++ Code SERIES NEW
Duration: 55 minutes

This series provides C++ developers with the knowledge and skills required to mitigate memory corruption vulnerabilities, protect data in transit using strong TLS ciphers, and to protect data using cryptographic best practices

---

COD 206

Creating Secure C++ Code NEW
Duration: 15 minutes

This course highlights some of the most useful security features for avoiding memory corruption vulnerabilities in C++, including:

- Using standard containers and their built-in functions to avoid direct memory operations.
- Using bounds-checking functions, especially for string manipulation, to avoid buffer overflows.
- Using smart pointers to avoid memory leaks associated with managing raw pointers.
- Using standard concurrency features to help reduce the risk of introducing race conditions.
- Using object-oriented programming features to define and manipulate data in terms of objects, thus avoiding direct memory operations that may lead to memory corruption.
- Using range-based loops to avoid off-by-one indexing errors.
- Using native regular expressions to validate untrusted text input and avoid the risk of introducing vulnerabilities through third-party libraries.

COD 207

Communication Security in C++
NEW Duration: 15 minutes

This course discusses hot to protect data in transit using encryption libraries and strong TLS ciphers. It also reviews important issues about public key certificates including signing and verifying them. After completing this course, you will be able to identify well-trusted encryption libraries and strong TLS cipher suites to protect data in transit, and explain how to protect and verify the integrity of public key certificates.

COD 307

Protecting Data in C++ NEW

Duration: 25 minutes

This course discusses cryptography and related issues for COD 307 - Protecting Data in C++.After completing this course, you will be able to generate strong encryption keys and identify related symmetric cryptography issues, such as pseudo random number generators (PRNGs), key derivation algorithms, and initialization vectors.Additionally, you will be able to select an appropriate symmetric encryption algorithm, cipher mode, and authenticated encryption mode, and identify common libraries that support symmetric cryptography.You will also be able to identify key concepts of public key cryptography, explain how public and private key pairs work together both to encrypt and decrypt data for secure transfer and to create and verify digital signatures, and implement best practices to mitigate memory exposure vulnerabilities.

COD 215

Creating Secure Code - .NET Framework Foundations
Duration: 90 minutes

This course describes .NET 4 security features, including concepts such as Code Access Security (CAS) and .NET cryptographic technologies. In addition, this course will introduce you to security changes in .NET 4 including level 2 security transparency, the new sandboxing and permission model, introduction of conditional APTCA, and changes to evidence objects and collections. This course provides secure coding best practices that will enable students to build more secure applications in .NET 4.

COD 219

Creating Secure Code - SAP ABAP Foundations

Duration: 90 minutes

This course presents best practices and techniques for secure SAP application development using Java and ABAP. It discusses basic application security principles, input validation in SAP applications, common application security vulnerabilities and mitigations, protecting data using encryption, and conducting security code analysis and code reviews.

COD 222

PCI DSS v3.2 Best Practices for Developers

Duration: 60 minutes

The Payment Card Industry Data Security Standard (PCI-DSS) Version 3.2 provides minimum requirements for addressing the security of software systems handling credit card information. Addressing the requirements during the design and build stages of the development lifecycle improves application security and simplifies compliance. This course will provide software developers with an in-depth understanding of application security issues within the PCI-DSS Version 3.2 and best practices for addressing each requirement.

COD 224

IoT SPECIALIZATION SERIES NEW

Duration: 60 minutes

In this series, you will learn about the importance of integrating security into each stage of your IoT SDLC.

COD 225

Insecure IoT Web Interfaces NEW

Duration: 10 minutes

In this course, you will learn how to identify common threats to IoT web interfaces and apply best practices to mitigate these threats.

COD 226

Insecure IoT Authentication and Authorization NEW

Duration: 10 minutes

In this course, you will learn about how to implement secure authentication and authorization for Internet of Things (IoT) devices.

COD 227

Insecure IoT Network Services NEW

Duration: 10 minutes

In this course, you will learn about the vulnerabilities of Insecure Network Services within the context of the Internet of Things (IoT) devices, and best practices to protect network services on IoT devices.

COD 228

Insecure IoT Communications NEW

Duration: 10 minutes

In this course, you will learn about the risks of insecure communications.

COD 229

Insecure IoT Mobile Interface NEW

Duration: 10 minutes

In this course, you will learn about best practices for protecting mobile applications used for IoT solutions

COD 230
Insecure IoT Firmware NEW
Duration: 10 minutes

In this course, you will learn how to securely distribute updates that fix known vulnerabilities in software or firmware for your Internet of Things devices.

COD 233
OWASP MOBLE SERIES NEW
Duration: 80

In this series, you will learn about the importance of integrating security into each stage of your Mobile App Development SDLC.

COD 234
Mobile Threats and Mitigations NEW
Duration: 20 minutes

In this course, you will learn about best practices for identifying and mitigating the most common threats to mobile applications and their data.

COD 235
Defending Mobile Data with Cryptography NEW
Duration: 20 minutes

In this course, you will learn about best practices for implementing strong cryptography to protect mobile applications and their data.

COD 236
Mobile App Authentication and Authorization NEW
Duration: 20 minutes

In this course, you will learn how to integrate secure authentication and authorization into your mobile application.

COD 237
Defending Mobile App Code NEW
Duration: 20 minutes

In this course, you will learn about best practices for defending your mobile application's code from attacks.

COD 242
Creating Secure SQL Applications
Duration: 40 minutes

In this course, you will learn how to protect sensitive data and while ensuring the integrity of applications running on the Microsoft SQL Server Engine and Azure SQL Database.

COD 251
Creating Secure AJAX Code - ASP.NET Foundations
Duration: 90 minutes

This course introduces secure ASP.NET coding principles for AJAX applications. It provides an overview of best practices to mitigate common vulnerabilities and protect against common attack vectors. Upon completion of this class, participants will be able to identify the threats to AJAX applications from cross-site scripting, cross-site request forgery, and injection attacks, and ways to implement countermeasures against these attacks by protecting client resources, validating input, protecting web services requests, preventing request forgeries, and securing data access.

COD 252
Creating Secure AJAX Code - Java Foundations
Duration: 35 minutes

This course introduces secure Java coding principles for AJAX applications. It provides an overview of best practices to mitigate common vulnerabilities and protect against common attack vectors. Upon completion of this class, participants will be able to identify the most common threats to AJAX applications from cross-site scripting, cross-site request forgery, and injection attacks, and ways to implement countermeasures against attacks by protecting client resources, validating input, restricting access to Ajax services, and preventing request forgeries.

COD 253
Creating Secure AWS Cloud Applications
Duration: 60 minutes

This course examines the security vulnerabilities, threats, and mitigations for AWS cloud computing services. It includes coverage of dedicated AWS security features, such as Key Management Service (KMS), Hardware Security Module (HSM), Identity and Access Management (IAM), and CloudWatch. In addition, it discusses how to leverage security features built into Common Amazon Cloud services, such as Simple Storage Service (S3), Elastic Compute Cloud (Amazon EC2), Elastic Block Store (EBS), Amazon Glacier, Relational Database Service (RDS), DynamoDB, Elastic MapReduce (EMR) , and Amazon Machine Images (AMI).

COD 254
Creating Secure Azure Applications
Duration: 90 minutes

This course examines the security vulnerabilities, threats, and mitigations for Azure cloud computing services. After completing this course, you will be able to identify the most common security threats to cloud based applications and best practices to protect against these threats. You will also be able to identify key Azure security platforms and services that you can use to improve the security of your applications.

COD 255
Creating Secure Code - Web API Foundations
Duration: 120 minutes

This course introduces the fundamentals of secure web services development. It describes common web services threats that might put your application at risk, and reviews best practices that you should incorporate to mitigate the risks from web services attacks. After completing this course, you will be able to describe various web services threats, explain the cause and impact of web services attacks, and implement secure development best practices to help protect web services.

COD 256
Creating Secure Code - Ruby on Rail Foundations
Duration: 90 minutes

In this course, you will learn about best practices and techniques for secure application development with Ruby on Rails. After completing this course, you will be able to identify and mitigate injection vulnerabilities, such as SQL injection and cross-site scripting, build strong session management into your Rails applications, and prevent other common vulnerabilities, such as cross-site request forgery and direct object access.

COD 257
Creating Secure Python Web Applications
Duration: 45 minutes

In this course, you will learn about best practices and techniques for secure application development with Python.After completing this course, you will be able to understand various types of injection vulnerabilities, including SQL injection and cross-site scripting. You will also be able to understand how to build strong session management into your Python web applications and how to prevent common vulnerabilities, such as cross-site request forgery, direct object access, and others. Finally, you will be able to recognize file system threats to web applications, including vulnerabilities with path traversal, temporary files, and insecure client redirects.

COD 260
SECURE SCRIPTING SERIES NEW
Duration: 120 minutes

In this series, you will learn about how to identify security threats to scripts and how to mitigate those threats by implementing access controls and following secure scripting best practices.

COD 261
***Threats to Scripts* NEW**
Duration: 30 minutes

In this course, you will learn about the impact of incorrect script development or lax security measures. You will also learn about the most common scripting vulnerabilities, including cached secrets, a variety of injection vulnerabilities, weaknesses related to permissions and privileges, and the threat of resource exhaustion.

COD 262
Fundamentals of Secure Scripting NEW
Duration: 30 minutes

In this course, you will learn about how shell scripting languages compare with more modern interpreted languages, several information security principles including least privilege and defense in depth, the importance of data validation, and operating system portability issues.

COD 263
Secure Scripting with Perl, Python, Bash and Ruby NEW
Duration: 30 minutes

In this course, you will learn about the importance of error and exception handling in shell scripts and interpreted languages, common syntax pitfalls, and how to prevent or mitigate several common vulnerabilities.

COD 264
Protecting Sensitive Data while Scripting NEW
Duration: 30 minutes

In this course, you will learn about how to use filesystem operations safely to protect files, techniques for system hardening, cryptography basics, and the importance of up-to-date communication security techniques.

COD 270
Creating Secure COBOL and Mainframe Applications
Duration: 25 minutes

This course covers countermeasures for security vulnerabilities on the mainframe, such as input validation, parameterized APIs, strong cryptography, and being aware of memory management issues

---

COD 280
Creating Secure Java Code Series NEW
Duration: 70 minutes

This series provides Java developers with the knowledge and skills required to implement the Java Security Model, JAAS, and to protect data using cryptographic best practices.

COD 281

Java Security Model NEW
Duration: 20 minutes

In this course, you will be introduced to Java's policy-driven security model. Key topics include the Java Security model, the Java security manager, security policies, and security policy files. After completing this course, you will be able to identify the components of the Java Security model and the functionality of the Java security manager and access controller. You will also be able to identify the components of Java security policies as well as describe the function of Java security policy files.

COD 282

Java Authentication and Authorization (JAAS) NEW
Duration: 20 minutes

This course discusses the Java authentication and authorization service, or JAAS. JAAS is a Java implementation of the standard pluggable authentication module, or PAM, framework. JAAS provides a framework that developers can use to require users to log in and to define precisely which actions users can perform. After completing this course, you will be able to identify the components of the JAAS framework, and identify how to use JAAS to control user authentications and authroization in your Java application.

COD 283

Java Cryptography NEW
Duration: 30 minutes

This course discusses cryptography and related issues in Java. After completing this course, you will be able to generate secure encryption keys and identify related issues such as pseudo random number generators, key derivation functions, and initialization vectors. You will also be able to select an appropriate symmetric encryption algorithm, cipher mode, and authenticated encryption mode. You will also be able to identify key concepts of public key cryptography, explain how public and private key pairs work together to encrypt and decrypt data for secure transfer and to create and verify digital signatures, and use the Java key tool command-line utility for creating and managing keys and keystores.

COD 292

IoT Embedded Systems Security - C/C++ Foundations
Duration: 30 minutes

This course module provides additional training on C/C++ Foundations of particular importance to embedded software engineers. The module contains the following features: Mapping of content to specific compliance and regulatory requirements Links to key reference resources that support the topics covered in the module "Knowledge Check" quiz that assesses mastery of key concepts.

COD 300

Protecting C Code Series NEW
Duration: 95 minutes

This series provides C developers with the knowledge and skills required to mitigate buffer overflow conditions, implement secure memory management best practices, and protect applications and data from attacks.

COD 301

Secure C Buffer Overflow Mitigations NEW
Duration: 45 minutes

The C and C++ languages cover a wide range of systems spanning several decades of development.Although all programming languages are susceptible to security vulnerabilities, C and C++ are particularly prone to them due to the low-level nature of the language.In this course, you will learn how to prevent the most serious vulnerabilities in your C and C++ applications.After completing this course, you will be able to mitigate buffer overflows, understand and prevent several additional types of memory management vulnerabilities, protect data in memory, prevent format string vulnerabilities, understand integer overflows, mitigate race conditions, and avoid the most common types of Injection vulnerabilities.

COD 302

Secure C Memory Management NEW
Duration: 30 minutes

After completing this course, you will able to identify the key concepts of dynamic memory management, identify common mistakes that lead to memory corruption and vulnerabilities, and implement best practices to mitigate memory management vulnerabilities

---

COD 303

Common C Vulnerabilities and Attacks NEW
Duration: 20 minutes

In this course you will review common C application vulnerabilities, how they manifest in code, and techniques and libraries that you can use to mitigate the risk of attack. After completing this course, you will be able to mitigate risk from format string attacks, integer overflows, race conditions, canonicalization issues, command injection, and SQL Injection.

COD 311

Creating Secure Code ASP.NET MVC Applications
Duration: 90 minutes

In this course, you will learn about ASP.NET MVC and Web API code security issues that affect MVC and Web API applications. You'll learn methods to protect your application from attacks against MVC's model-binding behavior, as well as methods to protect your application from cross-site scripting, cross-site request forgery, and malicious URL redirects. You will also study the Web API pipeline and how to implement authentication and authorization in Web API applications.

COD 312

Creating Secure C/C++ Code
Duration: 90 minutes

In this course, you will learn techniques for securing your C/C++ applications. You will learn about secure memory management in C/C++, protecting and authenticating sensitive data with symmetric and public key cryptography, and secure communications with TLS.

COD 313

Creating Secure Java Code
Duration: 60 minutes

In this course, you will learn how to identify and mitigate the most common Java code security vulnerabilities such as Injection, Overflows, Cross-Site Scripting and Information Disclosure. This course also describes how to use symmetric and asymmetric cryptography to protect data and applications in Java.

COD 314

Creating Secure C# Code
Duration: 90 minutes

This course describes methods to produce secure C# applications. It presents common security vulnerabilities that can be mitigated by proper input validation, other common security vulnerabilities and their mitigations, secure error handling and logging, and secure communication. The course also discusses unique features of C# and the .NET Framework that help protect against security vulnerabilities.

COD 315

Creating Secure PHP Code
Duration: 120 minutes

This course teaches PHP programmers the security principles they need to know to build secure PHP applications. This class teaches programming principles for security in PHP such as proper session management, error handling, authentication, authorization, data storage, use of encryption and defensive programming as well as avoiding and mitigating vulnerabilities such as SQL Injections, Cross-Site Scripting (XSS), File Inclusion, Command Injection, Cross Site Request Forgery (CSRF) and Null Byte attacks. With interactive knowledge checks in each of the modules, after completing the course, the student will be able to program securely and defensively in PHP.

COD 317
Creating Secure iOS Code in Swift
Duration: 90 minutes

In this course you will learn how to identify the most common iOS application security vulnerabilities, including Insecure Data Storage, Side Channel Data Leakage, Client Side Injection, Custom URL Scheme Abuse, Stack Smashing and Self-Signed Certificates. You will learn how to mitigate these threats by leveraging iOS and Swift security services while also implementing secure coding best practices, including Secure Memory Management, Automatic Reference Counting, Enabling Position Independent Executable, Secure Data Storage, Communicating Over HTTPS, App Transport Security, TLS Certificate Pinning, Asymmetric Encryption, Parameterized SQL Queries, Validating Path Location Input and Implementing Apple Pay.

COD 318
Creating Secure Android Code in Java
Duration: 90 minutes

In this course you will learn how to identify and mitigate the most common Android application security vulnerabilities and attack vectors, including: Weak Server Side Controls, Threats to Data, SQL Injection, Cross-Site Scripting (XSS), Session Hijacking, Threats to User Privacy and Confidentiality, Native Code Attacks, and Missing Data Encryption. Mitigation and best-practices include the Android software stack, the Android security model, access control methods, sandboxing, interprocess communications and implementing the security features of open-source developer tools.

COD 351
Creating Secure HTML5 Code
Duration: 90 minutes

This course provides in depth coverage on how to mitigate the most dangerous threats to HTML5 applications. It includes coverage of HTML5 Forms, WebSocket API, Server-Sent Events (SSE), Node.js security, jQuery security, the GPS API, static code analysis, and security packages. Upon completion of this class you will be able to identify key threats to your HTML5 application and then mitigate those threats by (1) leveraging built-in HTML5 security features and (2) implementing secure coding best practices.

COD 352
Creating Secure jQuery Code
Duration: 90 minutes

In this course, you will learn about common client-side vulnerabilities and threats to jQuery applications, and techniques for mitigating these vulnerabilities and threats. You will also learn about how to implement new HTML5 security features to secure JQuery applications, and best practices to secure local storage and implement transport layer security. After completing this course, you will be able to describe the threats that can impact your jQuery code and describe the countermeasures to address these threats.

COD 310
Protecting Java Code Series NEW Duration: 65 minutes

This series provides Java developers with the knowledge and skills required to mitigate the most common application security vulnerabilities, including SQLi, XSS, and Information Disclosure.

COD 380
Protecting Java Code : SQLi and Integer Overflows NEW
Duration: 10 minutes

This course describes ways to remediate common application security vulnerabilities in your Java application. After completing this course, you will be able to mitigate risk from SQL injection and integer overflows.

COD 381

Protecting Java Code: Canonicalization, Information Disclosure and TOCTOU NEW

Duration: 25 minutes

This course describes ways to remediate common application security vulnerabilities in your Java application. After completing this course, you will be able to mitigate risk from canonicalization issues, information disclosure, and race conditions.

COD 380

Protecting Data in Java NEW

Duration: 1 . minutes

This course describes ways to remediate common application security vulnerabilities in your Java application. After completing this course, you will be able to mitigate risk from canonicalization issues, information disclosure, and race conditions.

COD 392

IoT Embedded Systems Security - Creating Secure C/C++ Code

Duration: 30 minutes

This course module is a supplement to the Security Innovation course "Creating Secure C/C++ Code". It provides additional coverage on security topics that may be of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

COD 411

Integer Overflows - Attacks & Countermeasures

Duration: 60 minutes

An integer overflow is a programming error that can severely impact a computer system's security. Due to the subtlety of this bug, integer overflows are often overlooked during development. This course covers the security concepts, testing techniques, and best practices that will enable students to develop robust applications that are secure against integer overflow vulnerabilities.

COD 412

Buffer Overflows - Attacks & Countermeasures

Duration: 120 minutes

This course provides all the required information to understand, avoid and mitigate the risks posed by buffer overflows. The students are first provided with a detailed background on the mechanisms of exploit of stack-based and heap-based buffer overflows. The course then delves into the protections provided by the Microsoft compiler and the Windows operating system, such as the /GS flag and Address Space Layout Randomization (ASLR), followed by practical advice on how to avoid buffer overflows during the design, development, and verification phases of the software development life cycle. Practical examples are provided throughout the course to help students understand and defend against buffer overflows.

DES 101

Fundamentals of Secure Architecture

Duration: 60 minutes

In the past, software applications were created with little thought to the importance of security. In recent times, businesses have become more rigorous about how they buy software. When looking at applications and solutions, companies don't just look at features, functionality, and ease of use. They focus on the total cost of ownership (TCO) of what they purchase. Security is a large and visible part of the TCO equation. In this course, students will examine the state of the industry from a security perspective. They will then look at some of the biggest security disasters in software design and what lessons can be learned from them. Finally, participants will understand and use confidentiality, integrity, and availability as the three main tenets of information security. Upon completion of this course, participants will understand the state of the software industry with respect to security by learning from past software security errors and will avoid repeating those mistakes, and they will understand and use confidentiality, integrity, and availability (CIA) as the three main tenets of information security.

DES 201

Fundamentals of Cryptography

Duration: 120 minutes

In this course, you will learn basic concepts of cryptography and common ways that it is applied, from the perspective of application development. You will learn the importance of randomness; the roles of encoding, encryption, and hashing; the concepts of symmetric and asymmetric encryption; the purpose of cryptographic keys; and the roles of message authentication codes (MACs) and digital signatures. In addition, you'll be introduced to key management, digital certificates, and the public key infrastructure (PKI). Most importantly, you'll understand that cryptography is extremely complex, and requires strong expertise to be properly implemented and validated.

DES 212

Architecture Risk Analysis and Remediation

Duration: 60 minutes

This course defines concepts, methods, and techniques for analyzing the architecture and design of a software system for security flaws. Special attention is given to analysis of security issues in existing applications; however, the principles and techniques are applicable to systems under development. Techniques include accurately capturing application architecture, threat modeling with attack trees, attack pattern analysis, and enumeration of trust boundaries.

DES 213

SECURE ENTERPRISE INFRASTRUCTURE SERIES

Duration: 90 minutes

In this series, you will learn about the importance of designing and implementing secure access controls across the enterprise infrastructure. You will also learn about the techniques used to identify system security and performance requirements, develop appropriate security architecture, select the correct mitigations, and develop policies that can ensure the secure operation of your systems.

DES 214

Securing Network Access

Duration: 30 minutes

In this course, you will learn about how Network Access Control can be used to secure systems on a network, including how to integrate strong authentication, and how to enforce policies, and how using a centralized monitoring system to log system access and system use can be useful in recognizing and containing attacks.

DES 215
Securing Operating System Access
Duration: 30 minutes

This course equips employees to recognize the importance of understanding what constitutes private data and how to behave in a proactive manner to protect this information in their everyday work. In this course, you will learn about common operating system threats and how to best mitigate those threats. It also describes the benefits of multi-factor authentication, strong password management, and user account controls, and explains the benefits and risks associated with secure ID tokens, biometrics, and single sign-on (SSO).

DES 216
Securing Cloud Instances
Duration: 30 minutes

In this course, you will learn about the top threats to Cloud resources and how to mitigate them using application security best practices.

DES 217
Application, Technical and Physical Access Controls
Duration: 30 minutes

In this course, you will learn about the risks associated with data breaches and how to implement strong access controls and security policies that protect applications, systems and sensitive data.

DES 221
OWASP 2017 SERIES
Duration: 120 minutes

The primary objective of this series of courses, and of the OWASP Top 10, is to educated developers, designers, architects, managers, and organizations about the consequences of the most common and most important web application security weaknesses.

DES 222
Mitigating Injection
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with injection.

DES 223
Mitigating Broken Authentication
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with broken authentication.

DES 224
Mitigating Sensitive Data Exposure
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with sensitive data exposure.

DES 225
Mitigating XML External Entities (XXE)
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with XML External Entities (XXE).

DES 226
Mitigating Broken Access Control
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with broken access control.

DES 227
Mitigating Security Misconfiguration
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with security misconfiguration.

DES 228
Mitigating Cross Site Scripting (XSS)
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with Cross-Site Scripting (XSS).

DES 229
Mitigating Insecure Deserialization
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with insecure deserialization.

DES 230
Mitigating Use of Components with Known Vulnerabilities
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with using components with known vulnerabilities.

DES 231
Mitigating Insufficient Logging & Monitoring Vulnerabiliites
Duration: 12 minutes

In this course, you will learn how to mitigate the risks associated with insufficient logging and monitoring.

DES 292
Architecture Risk Analysis & Remediation for IoT Embedded Systems
Duration: 30 minutes

This course module provides additional Architecture Risk Analysis and Remediation training of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

DES 311

Creating Secure Application Architecture

Duration: 120 minutes

This course covers a set of key security principles that students can use to improve the security of application architecture and design. Principles of this course include applying defense to harden applications and make them more difficult for intruders to breach, reducing the amount of damage an attacker can accomplish, compartmentalizing to reduce the impact of exploits, using centralized input and data validation to protect applications from malicious input, and reducing the risk in error code paths.

DES 352

Creating Secure Over the Air (OTA) Automotive System Updates

Duration: 90 minutes

In this course, you will learn about the secure design considerations for over-the-air (OTA) updates for automotive systems. After completing this course, you will be able to identify the benefits and risks of OTA automotive system updates, understand the importance of public key cryptography to the security of these updates, and identify secure design considerations for development, delivery, and installation of OTA automotive system updates.

DES 391

Creating Secure Application Architecture for IoT Embedded Systems

Duration: 30 minutes

This course module provides additional training on Creating Secure Application Architecture of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

ENG 105
How to Integrate the Microsoft MS SDL into your SDLC
Duration: 90 minutes

This course introduces the fundamentals of the Microsoft Security Development Lifecycle (SDL) process. It covers the security requirements for each phase your SDLC, including: Requirements, Design, Implementation, Verification, and Release. It also includes coverage of the Agile SDL variation, the Security Development Lifecycle for Line-of-Business Applications (SDL-LOB), and the Microsoft SDL Threat Modeling tool.

ENG 205
Fundamentals of Threat Modeling *NEW*
Durations: 60 minutes

In this course, you will learn how to question-driven approach to threat modeling that can help you identify security design problems early in the application design process.

ENG 211
How to Create Application Security Design Requirements
Duration: 60 minutes

Security is an important component of an application's quality. To preserve the confidentiality, integrity, and availability of application data, software applications must be engineered with security in mind beginning with the design phase. Without defined security requirements, design choices will be made without security guidance and security testing cannot be effective. This course provides technical and non-technical personnel with the tools to understand, create and articulate security requirements as part of a software requirement documents. In this course, students will learn to apply the application security maturity (ASM) model to the development process, understand the security-engineering process, and describe the key security-engineering activities to integrate security in the development life cycle. Students will also be able to determine software security objectives, apply security design guidelines, and create threat models that identify threats, attacks, vulnerabilities, and countermeasures, in addition to learning to conduct security architecture and design reviews that help identify potential security problems, and minimize the application's attack surface.

ENG 301
How to Create an Application Security Threat Model
Duration: 90 minutes

Building secure software begins with creating a threat model to understand the potential threats to an application. The threat modeling process starts by asking what an attacker's goals might be, what information would be valuable to an attacker, and how would an attacker go about gaining access to that information? In this course, students will learn to identify the goals of threat modeling and the corresponding Software Development Lifecycle (SDLC) requirements, identify the roles and responsibilities involved in the threat modeling process, recognize when and what to threat model, and identify the tools that help with threat modeling. Students will learn to use the threat modeling process to accurately identify, mitigate, and validate threats.

ENG 311
Attack Surface Analysis & Reduction
Duration: 60 minutes

Attack surface analysis and reduction is an exercise in risk reduction. The attack surface of an application represents the number of entry points exposed to a potential attacker of the software. The larger the attack surface, the larger the set of methods that can be used by an adversary to attack. The smaller the attack surface, the smaller the chance of an attacker finding a vulnerability and the lower the risk of a high impact exploit in the system. This course provides an understanding of the goals and methodologies of attackers, identification of attack vectors, and how to minimize the attack surface of an application. In this course, students will learn to define the attack surface of an application, and how to reduce the risk to an application by minimizing the application's attack surface.

ENG 312

How to Perform a Security Code Review

Duration: 60 minutes

Application developers may use a variety of tools to identify flaws in their software. Many of these tools, however, cannot be deployed until late in the development lifecycle; dynamic analysis tools require a staging site and sample data, and some static analysis tools require a compiled build. Manual code reviews, in contrast, can begin at any time and require no specialized tools - only secure coding knowledge. Manual code reviews can also be laborious if every line of source code is reviewed. This course provides students with guidance on how to best organize code reviews, prioritize those code segments that will be reviewed, best practices for reviewing source code and maximize security resources.

ENG 352

How to Create an Automotive Systems Threat Model

Duration: 90 minutes

In this course, you will learn about the secure design considerations for over-the-air (OTA) updates or automotive systems. After completing this course, you will be able to identify the benefits and risks of OTA automotive system updates, understand the importance of public key cryptography to the security of these updates, and identify secure design considerations for development, delivery, and installation of OTA automotive system updates.

ENG 391

Create an Application Security Threat Model for IoT Embedded Systems

Duration: 30 minutes

This course module provides additional training on How to Create an Application Security Threat Model of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

ENG 392

Attack Surface Analysis and Reduction for IoT Embedded Systems

Duration: 30 minutes

This course module provides additional training on Attack Surface Analysis and Reduction of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

ENG 393

How to Perform a Security Code Review for ioT Embedded Systems

Duration: 30 minutes

This course module provides additional training on Performing Security Code Reviews of particular importance to embedded software engineers. The module contains the following features: Mapping of content to specific compliance and regulatory requirements Links to key reference resources that support the topics covered in the module "Knowledge Check" quiz that assesses mastery of key concepts.

TST 101
Fundamentals of Security Testing
Duration: 120 minutes

This course introduces security-testing concepts and processes that will help students analyze an application from a security perspective and to conduct effective security testing. The course focuses on the different categories of security vulnerabilities and the various testing approaches that target these classes of vulnerabilities. Several manual and automated testing techniques are presented which will help identify common security issues during testing and uncover security vulnerabilities.

TST 191
Fundamentals of Security Testing for IoT Embedded Systems
Duration: 30 minutes

This course module provides additional Fundamentals of Security Testing training of particular importance to embedded software engineers. The module contains the following features: Mapping of content to specific compliance and regulatory requirements Links to key reference resources that support the topics covered in the module "Knowledge Check" quiz that assesses mastery of key concepts.

TST 201
Testing for CWE SANS Top 25 Software Errors
Duration: 60 minutes

In this course, you will learn how to identify and mitigate each of the CWE's 25 Most Dangerous Software Errors. Coverage includes techniques for spotting common security issues through code review and testing. Secure coding best practices are included for each security defect, as well as descriptions of technology specific weaknesses. Upon completion of this course, you will be able to identify common security defects and their potential impact to your application. You will also be able to identify specific types of security vulnerabilities associated with different technologies. Finally, you will be able to apply the steps necessary to avoid, detect, and mitigate common types of security defects in your applications. The course includes Knowledge Checks, Module Summaries, and information about additional online resources.

TST 291
Classes of Security Defects - IoT Embedded Systems
Duration: 30 minutes

This course module provides additional training on Security Defects Classes of particular importance to embedded software engineers. It includes mapping of content to specific compliance and regulatory requirements, links to key reference resources that support the topics covered in the module, and a "Knowledge Check" quiz that assesses mastery of key concepts.

TST 211
How to Test for the OWASP Top 10
Duration: 90 minutes

The Open Web Application Security Project (OWASP) Top Ten is a listing of critical security flaws found in web applications. Organizations that address these flaws greatly reduce the risk of a web application being compromised, and testing for these flaws is a requirement of the Payment Card Industry Standards (PCI-DSS) as well as other regulatory bodies. This course explains how these flaws occur and provides testing strategies to identify the flaws in web applications.

TST 401

Advanced Software Security Testing - Tools & Techniques

Duration: 120 minutes

This course delves deeply into the techniques for testing specific security weaknesses. The class is broken down into the three areas where bugs are most often found: insecure interaction between components, risky resource management, and poor defenses. Tools and techniques for security testing are presented, including ten different types of attacks such as SQL Injection, Command Injection, Cross-site Scripting, Buffer Overflow and Access Spoofing. After taking this course, the student will be able to understand the ten types of attacks; know which tools to use to test for these attacks; test software applications for susceptibility to the ten specific attacks; describe the expected mitigations required to prevent these attacks.

TST 411

Exploiting Buffer Overflows

Duration: 120 minutes

This course provides students with all the required information to help understand and mitigate buffer-overflow exploits. It first introduces the concepts necessary to recognize the threats posed by these exploits, and to comprehend the mechanisms behind exploitation of stack-based and heap-based buffer overflows. The course then delves into the different challenges faced by exploit code and how different exploitation techniques overcome environmental limitations.

TST 491

IoT Advanced Embedded Software Security Testing

Duration: 30 minutes

This course module provides additional Software Security Testing of particular importance to embedded software engineers. The module contains the following features: Mapping of content to specific compliance and regulatory requirements Links to key reference resources that support the topics covered in the module "Knowledge Check" quiz that assesses mastery of key concepts.

ENG 110
Essential Account Management Security - NEW
Duration: 15 minutes

This course provides essential guidance to information system managers, designers and program managers on implementing specific account management security controls at the hardware and software level to facilitate compliance with applicable regulatory requirements.

ENG 111
Essential Session Management Security - NEW
Duration: 15 minutes

This course provides essential guidance to system designers and developers on implementing specific session management security controls at the software level to facilitate compliance with applicable regulatory requirements.

ENG 112
Essential Access Control for Mobile Devices - NEW
Duration: 15 minutes

This course provides essential guidance to mobile system designers and developers on implementing technical controls at the software and device level to facilitate compliance with applicable regulatory requirements.

ENG 113
Essential Secure Configuration Management - NEW
Duration: 15 minutes

This course provides essential guidance to program managers, system designers and developers responsible for the effective implementation of selected security controls and control enhancements to help ensure compliance with applicable regulatory requirements.

ENG 114
Essential Risk Assessment - NEW
Duration: 15 minutes

This course provides essential guidance to individuals with information system, security, and/or risk management and oversight responsibilities that include defining the purpose, scope, roles, management commitment, and coordination among organizational entities to help ensure compliance with applicable regulatory requirements.

ENG 115
Essential System and Information Integrity - NEW
Duration: 15 minutes

This course provides essential guidance to program managers, system designers and developers on identifying systems affected by software flaws, including potential vulnerabilities resulting from those flaws, and report this information to designated organizational personnel.

ENG 116
Essential Security Planning Policy and Procedures - NEW
Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing and disseminating an organization-wide security planning policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance.

ENG 117

Essential Information Security Program Planning - NEW

Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing and disseminating an organization-wide information security program plan to facilitate compliance with applicable regulatory requirements.

ENG 118

Essential Incident Response - NEW

Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for implementing an incident response policy and associated controls to help ensure compliance with applicable regulatory requirements.

ENG 119

Essential Security Audit and Accountability - NEW

Duration: 15 minutes

This course provides essential guidance to information system owners, system administrators, and information system security officers developing procedures to facilitate the implementation of the audit and accountability policy and controls to facilitate compliance with applicable regulatory requirements.

ENG 120

Essential Security Assessment and Authorization - NEW

Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing and implementing personnel security policy and associated personnel security controls to help ensure compliance with applicable regulatory requirements.

ENG 121

Essential Identification and Authentication - NEW

Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing identification and authentication policy and controls to help ensure compliance with applicable regulatory requirements.

ENG 122

Essential Physical and Environmental Protection - NEW

Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing physical and environmental protection policy and associated physical and environmental protection controls to help ensure compliance with applicable regulatory requirements.

ENG 123
Essential Security Engineering Principles - NEW
Duration: 15 minutes

This course provides essential guidance to program managers, system designers, developers, information security engineers and systems integrators responsible for applying security-engineering principles to new development information systems or systems undergoing major upgrades.

ENG 124
Essential Application Protection - NEW
Duration: 15 minutes

This course provides essential guidance to system designers and developers on implementing specific application security controls at the software level to facilitate compliance with applicable regulatory requirements.

ENG 125
Essential Data Protection - NEW
Duration: 15 minutes

This course provides essential guidance to information system managers, information security managers, system designers and developers on implementing cryptographic controls at the information systems level to facilitate compliance with applicable regulatory requirements.

ENG 126
Essential Security Maintenance Policies - NEW
Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing procedures to facilitate the implementation of the system maintenance policy and associated system maintenance controls.

ENG 127
Essential Media Protection - NEW
Duration: 15 minutes

This course provides essential guidance to individuals with information security implementation and operational responsibilities for developing and disseminating an organization-wide information media protection policy that addresses purpose, scope, roles, responsibilities, management commitment, and coordination among organizational entities to facilitate compliance with applicable regulatory requirements.

AWA 601

Information and Application Security Awareness

This course begins by describing the CIA (Confidentiality, Integrity, Availability) triad and specifically, what it means in an application security context. It then examines the root cause of software vulnerabilities, how attackers view your applications, the true cost of software vulnerabilities, and how to integrate security into your development and IT organizations. To illustrate real world situations, the instructor will conduct live demonstration exercises.

Upon completion of this class, participants will be able to understand:

- The difference between functional and security bug
- How applications are attacked, how to identify assets, entry points and attack vectors, and how to leverage tools and techniques to exploit vulnerabilities
- Threats to applications and countermeasures that can be applied during development to mitigate them

Modules Covered:

What is Security?

This module presents the CIA (Confidentiality, Integrity, Availability) model and discusses how to define security and secure systems.

Why Does Security Matter?

This module describes the reliance of businesses and critical systems on software and explores the consequences of failure.

Thinking Like an Attacker

This module presents the thought process and techniques that attackers use to break software applications.

Security and the Software Development Lifecycle (SDLC)

This module describes best practices for integrating security into the organization and building culture of security. It will also demonstrate that most security problems are not in security-specific components, but rather they are errors in general software routines and functions.

Case Studies

In this module, the instructor will walk students through recent data breaches where application security vulnerabilities have resulted in huge financial losses. More importantly, they will examine root causes and describe defensive countermeasures that could have helped to prevent and/or reduce the impact of the breaches. These studies look beyond IT disruption and into broader consequences such as impact on stock value, remediation expense, reputation loss, liability, etc.

Attacks and Defense

This module examines the threats that can be mitigated at the network layer, as opposed to those that must be addressed in software. This module also covers broad classes of attack tools (black box, white box, and gray box).

COD 715
Creating Secure Code - .NET (C#)

Secure coding is the process of reducing the susceptibility of .NET code to vulnerabilities. This course gives developers an in-depth immersion into secure coding practices, with an emphasis on the security features and pitfalls of the .NET programming environment. It also introduces the concept of Threat Modeling, which is a highly regarded risk mitigation technique to secure development. To complement the knowledge and techniques presented, this course includes hands-on labs on implementing secure solutions in .NET and real-world examples of how to find, fix and prevent vulnerabilities.

At the end of this course, participants will be able to:
- Identify common security issues and attack vectors in all applications
- Understand and implement secure design and development techniques
- Implement best practices for securely developing .NET applications and protecting data

Modules Covered:

Introduction
- The underlying cause of software vulnerabilities and the impact they can have on an organization
- The difficulties in integrating improved security in an environment with opposing goals
- Software failures, the requirement for security at the application layer, compliance issues, and the goals of a security improvement system

Common Coding Errors

Common Web Application Errors
- Common security vulnerabilities found in web applications and the risk they carry

Defensive Coding Principles
- Coding principles that help in the design and development of secure software
- How to apply principles correctly so that they provide a foundation for development standards and result in fewer vulnerabilities
- Pitfalls to avoid and methods that will prevent and remediate security vulnerabilities

Threat Modeling

Overview of the threat modeling process

COD 721
Attacker Techniques Exposed: Threats, Vulnerabilities, and Exploits

This course examines trends in software vulnerabilities, demonstrates examples of security breaches, explores a wide range of live software vulnerabilities, and introduces threat modeling techniques.

Upon completion of this class, participants will be able to:
- Recognize the need for integrating security at each phase of the Software Development Lifecycle
- Identify process gaps that are needed to improve the security of their systems
- Create a high-level map of needs for the organization's people, processes, and technology

Modules Covered:

The Potential Attacker

This module discusses the different genres of attackers, as well as their varying skill sets and goals.

The Anatomy of an Attack

This module examines the different steps of an attack, from information gathering to the attack's consequences.

Attacks and Defenses

This module provides an overview of the layered security model and the different defenses that will help mitigate security risks.

Live Vulnerability and Exploit Tour

Participants will be shown live examples of a wide array of vulnerabilities and exploits, providing awareness and key insight as to how an attacker views and exploits applications.

Tools and Threats

The overall threat to applications is growing and so is the number of tools that make it easier for hackers to exploit them. This module discusses the underground world of the attacker and the range of tools available to them.

Thinking like the Attacker: Threat Modeling

A critical step in securing an application or system is to methodically think through threats. This module presents several techniques for threat modeling, and describes the process of modeling threats against several systems.

Incorporating Threats into Software/System Design, Development, Testing and Deployment

By considering threats at each stage of the development lifecycle, development teams can make more informed decisions to create software and systems that are more resilient to attack. This module covers tools and techniques for mitigating threats at each phase of development.

COD 722

PCI Bootcamp for Software Development Teams

This course introduces the PCI-DSS to those responsible for compliance in software development. The goals of software security, the impact of security on a business, and the difficulties in achieving perfect security are presented. The software requirements of the PCI-DSS and PA-DSS are broken out from the standards and presented clearly. Integration of security into the software development lifecycle and the form, methods, and remediation of the most common software vulnerabilities are shown.

Upon completion of the course, participants will be able to:

- Understand the software security landscape
- Realize the contribution of security to the total cost of ownership for software
- Understand why secure software can be difficult to achieve
- Know the fundamental requirements for the PCI-DSS and PA-DSS
- Integrate security into the software development lifecycle
- Conduct common attacks and think like an attacker
- Identify and remediate common vulnerabilities

Modules Covered:

Introduction to Software Security

- The fundamental tenets of software security and its importance in the business environment
- Compliance issues and the rise of compliance as a motivation for improving software security
- Why software security is challenging and the approaches needed to create secure software

Introduction to the PCI-DSS

Fundamentals of Security in the SDLC

- The software development lifecycle and the need to integrate security from beginning to end
- Creating secure requirements and design, secure coding, secure deployment, and post-deployment activities

Common Weaknesses and Vulnerabilities

- Attacks that carry the highest risk for a PCI compliant application and detailed information about the underlying causes and methods of exploit
- Methods for identifying and remediating vulnerabilities

COD 813
Creating Secure Code - Java

Secure coding is the process of reducing the susceptibility of code to vulnerabilities. This course gives developers an in-depth immersion into secure coding practices with an emphasis on the security features and pitfalls of the Java programming environment. To complement the knowledge and techniques presented, it includes hands-on labs on implementing secure solutions in Java and real-world examples of how to find, fix and prevent vulnerabilities.

Upon completion of this course, participants will be able to:
- Leverage Java security architecture and it's built-in security features to reduce application security risk
- Properly handle cryptography and permissions
- Avoid Java vulnerabilities by using Java coding best practices.
- Recognize and remediate common Java coding errors that lead to vulnerabilities
- Write defensive code that protects your application from common threats
- Understand the do's and don'ts of managed code
- Recognize when code is required to be reviewed for security vulnerabilities

Modules Covered:

Introduction
This module offers a "State of Software Security" address and why it's still woefully lacking. It introduces the concept of entry points, the primary means that software is exploited. Additionally, it will provide examples of when good design goes bad and a framework for how to think about software security (and how it's different from network security).

Threat Modeling
This module illustrates how threat modeling is leveraged to identify potential threats to your application, uncover and prioritize security vulnerabilities, and guide your secure programming and security testing efforts. It includes a Lab that entails conducting a threat model.

Common Coding and Design Errors
This module describes the top ten most common programming errors that lead to security vulnerabilities. Leveraging real-world examples, the instructor will show you what vulnerabilities look like in code, and how to leverage manual and automated techniques to find them. Most importantly, you'll learn how to remediate them and countermeasures to mitigate them.
Coding errors covered:
1. Trusting the identity of a remote host
2. Poorly implementing cryptography
3. Not validating user input
4. Information disclosure
5. Integer overflows
6. Relative and default paths
7. Administrative, software and service back doors
8. Storing sensitive data in plain text
9. Creating temporary files
10. Trusting libraries and OS APIs

Common Web Application Errors
This module describes how Web applications are different from other platforms and how they are typically attacked and examines the most common errors in Web Applications that lead to security vulnerabilities. For each vulnerability, the instructor will include examples of how to find the error, how to fix the error and how to leverage ASP.NET's built-in security protections can help (where applicable).
Errors covered:
1. Trusting Client-Side Validation
2. Cross Site Scripting
3. Command Injection
4. Forceful Browsing
5. Broken Authentication
6. Disclosing too much information

Defensive Coding Principles
This module focuses on 19 secure design and coding principles and provides in-depth examples of how to apply these principles

COD 817
Creating Secure Code - iOS

In this course, participants will learn to develop and deploy secure iPhone applications by leveraging Apple's security libraries and frameworks. Participants will also learn secure coding best practices for iOS and how to properly deploy iOS and Xcode security features.

Upon completion of this course, participants will be able to:
- Apply mobile development best practices in coding
- Use the security features in iOS to improve the security of mobile applications
- Be able to identify and remediate common mobile security vulnerabilities
- Apply iOS security best practices in development
- Deploy security services available in iOS
- Identify iPhone application security risks
- Understand the role of Apple iOS and SDK tools in providing security to iPhone applications

Modules Covered:
Mobile Application Development Best Practices
This module presents proven best practices and tools for improving the security and privacy posture of their mobile applications.
Introduction to iOS Devices and System Security
This module describes the iOS Generic Security Services (GSS) framework and it's numerous security features, and how developers can leverage it to produce more secure code.
Common iOS Application Vulnerabilities, Threats and Mitigations
iPhone attack vectors include web-based malware, SQL injection, session hijacking, theft of data at rest and in transit, and jailbreaking. This module helps you understand iPhone security vulnerabilities and attack vectors so that you are able to implement key mitigation techniques during development.
Secure iOS Mobile Application Best Practices
This module provides language- and tool-specific instructions on how to integrate Apple security services into your own secure coding best practices to protect against critical vulnerabilities. It includes multiple hands-on labs that demonstrate defensive coding techniques to harden your iPhone applications. Topics covered:
- Deploying iOS and Xcode security features
- Build hardening
- Enabling Automatic Reference Counting (ARC)
- Enabling Position Independent Executable (PIE)
- Enabling Stack Protector

COD 818
Creating Secure Code - Android

Android applications must follow the same security principles as other applications, but developers can leverage built-in security libraries and other features to help prevent common application vulnerabilities. This course helps participants develop secure Android applications by applying Android-specific secure development techniques. It examines vulnerabilities that are specific to the Android platform and provides real-world examples (illustrated in code) of failures and methods to find, fix, and prevent each type of flaw. In the hands-on labs, participants will discover vulnerabilities for themselves and find ways to remediate or mitigate them, greatly enhancing the security of their code.

Upon completion of this course, participants will be able to:
- Identify common security issues and attack vectors in Android applications
- Identify security features of the Android OS, SDK, and NDK
- Understand application-based permissions, data protection methods, code signing, packaging, and updating techniques used to secure Android applications
- Implement best practices for securely developing Android applications and securing data

Modules Covered:
Mobile Application Development Best Practices
This module presents known best practices and tools for improving the security and privacy posture of mobile applications.
Introduction to Android System Security
Like most operating systems, Android is best visualized as a set of layers, with each layer supporting the ones above. The Android layers are Linux, Libraries and Runtime, Application Framework, and Applications. In this module, participants will learn how to integrate security services of Android's Linux kernel, SDK, and hardware into their applications.
Common Android Application Threats and Mitigations
In many cases, Android attack vectors are not unique to the mobile device. This module presents common web-based attacks, and provides an overview of the security features and vulnerabilities that are part of the Android environment.
Android Secure Development Best Practices
In this module, participants will learn how to protect their Android application by following secure coding best practices using Java. These principles will guide design and development, and reduce the frequency and severity of common vulnerabilities.

COD 892
Creating Secure Code - Embedded C/C++

Most embedded system software is developed in C/C++. It is a natural choice because of its portability across platforms, efficient use of system resources, and ability to interact directly with embedded operating systems such as Linux. However, the trade-off of these features is that embedded C/C++programming is challenging because system resources are scarce, your tools are limited, and the risks are high. Additionally, embedded C carries all the risk of large system C/C++ programming, often with less margin for error, and worse outcomes in the event the application is compromised.
This course examines coding errors and vulnerabilities in the context of embedded C/C++programming and provides detailed code examples of insecure practices and methods to find, fix and prevent each type of flaw. Participants are provided with a set of security coding best practices and practical recommendations.

At the end of this course, participants will be able to:
- Understand the embedded vulnerability landscape
- Proactively recognize and remediate coding errors that lead to vulnerabilities in embedded software
- Implement techniques for mitigating risk against vulnerabilities
- Perform threat modeling to identify vulnerabilities and analyze risk

Modules Covered:
The Embedded Security Landscape
Improper Neutralization of Special Elements
Buffer Copy without Checking Size of Input
Mitigating Buffer Overflow Conditions
Dangers of Uncontrolled Format Strings
Use of Hard-Coded Authentication Credentials and mitigation techniques
Reliance on Untrusted Inputs in a Security Decision
- Common Authentication Errors
- Mitigations

Validating that your Compiler is Set for Security
Insecure Coding Examples (detailed and specific to embedded C/C++)
Threat Modeling
- Collecting Information
- Decomposing the application
- Building the activity matrix
- Building the threat profile

DES 721
OWASP Top 10 Threats & Mitigations

This course introduces students to OWASP and the Top 10 Project, and covers in detail each of the OWASP Top 10 Web Application Vulnerabilities. The instructor will provide examples and demonstrations of the vulnerabilities and exploits, and describe remediation techniques and best practices to avoid them. Specific examples include the usage of built-in and 3rd party functions, and libraries for Java and .NET.The class will utilize "Super Secure Bank," a vulnerable web application developed by our experts to demonstrate and teach web application security issues. Student will have the opportunity to see the vulnerabilities in action, and personally perform live exploits against the application to reinforce the principles introduced.If the one-day option is selected, each OWASP Top 10 vulnerability will be covered, but will not include hands-on labs.

If the two-day option is selected, labs for most of the OWASP Top 10 are included. Particular focus will be placed upon the most common and serious OWASP Top 10 vulnerabilities, including:
- Cross-Site Scripting (XSS)
- SQL Injection (SQLi)
- Cross-Site Request Forgery (CSRF)
- Broken Authentication & Session Management

Modules Covered:
Introduction to Web Application Security
This module provides an overview of the OWASP and what the OWASP Top 10 means from a development perspective.
Testing Web Applications with a Proxy
The instructor will introduce Burp Web Proxy and discuss how it can be effectively leveraged to test common web applications.
Topics include:
OWASP Top 10 in Detail
For each Top 10 vulnerability, the instructor will describe the risk and impact, and how to detect and remediate the vulnerability.

DES 722
CWE/SANS Top 25 - Threats & Mitigations

This course covers in detail the CWE/SANS Top 25 Most Dangerous Programming Errors, which comprises weaknesses in all types of software applications: Web, Operating System, Mobile, Embedded, Desktop, etc. It is structured in a "teach then do" format, wherein it combines instructor-led training and hands-on labs, in which students implement what they learned with assistance from the instructor.The two day baseline class is designed for all software development stakeholders, and focuses on gaining a solid understanding of each of the Top 25 Most Dangerous Software Errors.There are two optional follow-up sessions - one for Developers and one for Testers:•The Developer-focused day recaps the Top 25 and goes into more detailed code examples of each vulnerability.•The Tester-focused day includes demonstrations and a hands-on lab where students are able to attack a vulnerable website using the Common Attack Patterns related to the Top 25. It also includes exercises on how to identify and exploit the vulnerabilities, as well as remediation recommendations.

Specific topics covered include:
- Detailed description of each of the Top 25 software weaknesses with demonstrations and code example
- Discussion of detection methods for the Top 25, including an introduction to tools such as Burp Proxy Suite
- Leveraging Threat modeling to better understand potential attack frequency and attacker methods
- Other weaknesses that did not make the Top 25 list but are important to know about
- Attack patterns for each weakness
- Upon completion of the two day course, students will be able to:
- Recognize the attributes and causes of each CWE/SANS dangerous programming error
- Understand the practices that help prevent the most common mistakes and lead to the prevention of CWE/SANS coding error
- Recognize how these software security defects/weaknesses can be exploited
- Apply testing techniques to discover weaknesses

Modules Covered:
The Top 25 Most Dangerous Software Errors
This module is the heart of the course and provides comprehensive details on each of the Top 25 including:
- What the error is, the risk it carries, and how an attacker can exploit it
- Testing considerations and how to detect the error
- Mitigations, countermeasures and common defenses

The Top 25 are divided into these categories to facilitate discussion and demonstration:
This module includes code samples of poorly written code, as well as real-world examples of exploit techniques. Additionally, the instructor will discuss briefly software errors that were considered for inclusion on the Top 25, but did not make it to the final list.
Optional Developer-focus Day
This module focuses on key software security development principles and presents six essential security-engineering practices that will help developers build more secure and robust applications.
Optional Tester-focus Day
This module starts with an overview of security test tools, and browser-based proxy for testing web applications. The remainder of the day is spent in a hands-on lab that allows students to conduct attacks on a vulnerable web application. The instructor will take the theoretical discussion from the previous days and turn it into a practical testing experience. This module covers the following topics:
- Reconnaissance
- Data Leakage/Information Exposure Issues
- Injection Attacks
- Modification of Assumed Immutable Data (MAID) Attacks
- Authentication Issues
- Poor Configuration

DES 811
Secure Architecture & Design

Architecture reviews are one of the most cost effective ways to discover security issues proactively in applications, yet only a small fraction of development teams understand how to successfully conduct an architecture review. This class addresses this gap by allowing students to use their own software application in lab-based exercises. By learning how to examine your own application's architecture for issues, your team can cost-effectively resolve key security issues before they make their way into your software application. Conversely, if your system is already in production, an architecture review can be a good way to identify major security gaps that may have been previously unknown.

Upon completion of this class, participants will be able to:
- Understand network security issues that may apply to an architecture
- Understand host security issues that may apply to an architecture
- Understand application security issues that may apply to an architecture
- Understand that managing security is a risk management exercise
- Learn how to break down an application's architecture and identify security flaws

Modules Covered:
Network Security Recommendations
This module explains how decisions about network architecture can affect the security of an application, and introduces topics such as network scanning, sniffing, and encrypted/unencrypted protocols.
Host Security Recommendations
This module presents recommendations for hardening hosts to protect running applications and covers malware, rootkits, and attacker techniques.
Application Security Recommendations
This module provides an overview of high-level application security strategies including defense in depth, input validation, and principle of least privilege.
Risk Ranking
This module helps participants understand how to determine the level of risk inherent with the applications they are currently working with. Participants will complete an exercise that quantifies the risk level of different aspects of an application.
Architecture Review
The instructor will lead participants through an architecture security review exercise where they map out and review familiar application architectures. This helps to illustrate how an architecture review will often lead to the discovery of serious issues.

ENG 801
Effective Threat Modeling

This course introduces the technique of threat modeling, its primary goals, and its role within software development. Once you are familiar with the concepts behind threat modeling, the entire threat modeling process is demonstrated – giving you the knowledge you need to apply threat modeling to your own products and design/develop more secure code.

Upon completion of this course, participants will be able to:
- Identify the goals of threat modeling
- Understand the importance of early lifecycle security best practices such as threat modeling
- Identify the roles and responsibilities involved in the threat modeling process
- Use Threat Modeling to accurately identify, mitigate, and validate threats
- Leverage various tools to assist in threat modeling
- Create a threat model of your software

Modules Covered:

Defining Threat Modeling
This module presents detailed information that will allow participants to understand the importance of threat modeling to mitigate risk, and quickly gain an understanding of how to approach building threat models of their software. Upon completion of this course, participants will be able to:
- Identify the goals of threat modeling
- Recognize the relationship between threat modeling and the Software Development Lifecycle (SDLC)
- Identify the roles involved in the threat modeling process
- Understand what and when to threat model

Applying Threat Modeling Process
This module describes in detail the threat modeling process and procedures to follow in order to apply each step. It includes a lab to help participants apply what they've learned in a real-world scenario. After completing this module, participants will be able to:

Optional: Advanced Threat Modeling 1/2 Day
If desired, we can add another half-day of training that provides more in-depth coverage of threat modeling principles and

ENG 812
Security Code Review

This course presents the primary techniques used to conduct a security code review, with the focus of identifying potential security vulnerabilities. Numerous exercises and examples of code are provided, along with guidance on how to efficiently identify areas that need a more in-depth review.

Upon completion of this course, participants will be able to:
- Implement discovery methods to uncover flaws in the source code
- Apply the knowledge of detecting security vulnerabilities to perform a successful security code review
- Use manual methods as well as automated tools to conduct source code reviews
- Leverage the results of code reviews to make improvements in the secure software development process

Modules Covered:

Introduction to Secure Code Review
This module introduces participants to key terms and how they are used in this course. The focus is on understanding the basic concepts behind a secure code review and the techniques required to perform a successful review.

Secure Code Review Methodology
This module presents the methodology used to perform a secure source code review, which includes identifying the types of issues to examine in the code, and then how to remediate these vulnerabilities as quickly and effectively as possible. Students will use threat models, architecture diagrams, and other inputs to guide the review, and then leverage the list of discovered vulnerabilities to guide future reviews.

Common Hotspots in Source Code Review
This module presents the common hotspots that are susceptible to vulnerabilities. The goal is to understand the various types of static code vulnerabilities, their impact, and how to detect them by reviewing the source code. Demonstrations and code snippets will be used to showcase these issues.

Application-specific Hotspots in Source Code Review
This module explains the various ways to detect business logic flaws in applications, and how to identify vulnerabilities that surface due to specific application features.

Post Code Review Activities
This module explains the various activities that should be performed once the code review phase has been completed. These activities range from bug prioritization to security knowledge transfer for the development team, so as to avoid repetition of these bugs in the future. The module concludes with a discussion of the various approaches that can be used to perform the

TST 901
Advanced Web Application Security Testing

Course Overview: This course is designed to take knowledgeable web application security testers into the realm of expertise - equipping them with knowledge of advanced web application testing techniques and penetration testing tools. Participants will learn about many important web vulnerabilities like HTML5 attacks, business logic attacks, web services attacks, and AJAX/JSON specific vulnerabilities and issues.

Upon completion of this course, participants will be able to:
- Build a security test plan, driven by threat modeling
- Efficiently and accurately identify the high-risk areas of an application
- Understand and apply sophisticated web application testing techniques
- Identify vulnerabilities in web applications that are unlikely to be found using automated tools

Modules Covered:
The OWASP Top 10 Review
This module recaps the OWASP 10 most critical web application vulnerabilities. This provides a good review of foundational information to help the student understand the vulnerabilities and the advanced testing techniques.
HTML5 Attacks
This module discusses HTML5 has tags, event handlers, and APIs that offer the attacker many more possible attack vectors. This module focuses on HTML5 attacks - web storage, cross-site scripting, clickjacking, cross origin resource sharing (CORS) attacks.
Business Logic Attacks
This module describes effective manual test techniques that need to be conducted to determine if your application is vulnerable to business logic attacks.
XML Attacks
This module presents a variety of XML attacks and techniques that testers can use to find the vulnerabilities.
Web Services Attacks
In this module, the instructor describes various types of web services attacks and countermeasures that can be implemented to mitigate them.
AJAX/JSON Attacks
This module covers parameter manipulation, injection attacks, and JavaScript high jacking.
Silverlight and Flash Attacks
This module focuses on techniques for attacking these technologies to determine if your rich client application is vulnerable.
In all of the modules, each attack is described and the techniques to verify the vulnerability are shown.