CertNexus Cyber Secure Coder Course Description:
Exam CSC-210

Course Content

*Orange text indicates activity

Lesson 1: Identifying the Need for Security in Your Software Projects
Topic A: Identify Security Requirements and Expectations
- Security Throughout the Development Process
- Business Requirements
- Standards and Compliance Requirements
- User Impact
- User Expectations
- Platform Requirements
- Consequences of Not Meeting Security Requirements
- Guidelines for Identifying Security Requirements and Expectations
- Identifying Security Requirements and Expectations

Topic B: Identify Factors That Undermine Software Security
- Three Ps of Software Security
- Software Security Terminology
- Identifying Factors That Undermine Security

Topic C: Find Vulnerabilities in Your Software
- Builders and Breakers
- Hacking
- Phases of an Attack
- Common Attack Patterns
- Case Study: Protecting Against a Password Attack
- Guidelines for Identifying Software Security Vulnerabilities
- Identifying Vulnerabilities in an Application
- Cracking a Password Hash
- Fixing a Password Hash Vulnerability

Topic D: Gather Intelligence on Vulnerabilities and Exploits
- Vulnerability Intelligence
- Exploits
- Guidelines for Researching Vulnerabilities and Exploits
- Identifying Sources for Vulnerability Intelligence

Lesson 2: Handling Vulnerabilities

Topic A: Handle Vulnerabilities Due to Software Defects and Misconfiguration

- Software Defects
- Causes of Software Defects
- Guidelines for Preventing Security Defects
- Preventing Security Defects
- Problems in Third-Party Code
- Problems in Standard Libraries
- Dependencies
- Encryption Validation
- Security of Host Systems and Service Providers
- Guidelines for Using Third-Party Code and Services
- Host Platform Configuration
- Hypervisor Vulnerabilities
- Guidelines for Managing Vulnerabilities in External Hosts and Services
- Identifying Vulnerabilities in a Software Project
- Examining the Project Files
- Error Messaging
- Error Handling
- Fail-Safe
- Failure Recovery
- Guidelines for Secure Error Handling
- Identifying Software Defects and Misconfiguration

Topic B: Handle Vulnerabilities Due to Human Factors

- The Human Element in Software Security
- Vulnerabilities Attributed to the Human Element
- Social Engineering Attacks
- User Input
- Input Validation
- Security Policy Enforcement
- Guidelines for Managing People Risks
- Managing People Risks

Topic C: Handle Vulnerabilities Due to Process Shortcomings

- Development Process Approaches
- Building Security In
- The CIA Triad
- Requirements Phase
- Design Phase
- Development Phase
- Testing Phase
- Security Testing Tools
- Deployment Phase
- Maintenance Phase
- Development Process Security
- Guidelines for Software Development Processes
- Managing Software Development Process Risks

Lesson 3: Designing for Security

Topic A: Apply General Principles for Secure Design

- Security in the Design Phase
- Security by Obscurity vs. Security by Design
- OWASP Security Design Principles
- Minimize Attack Surface Area
- Establish Secure Defaults
- Least Privilege
- Least Common Mechanism
- Defense in Depth
- Fail Securely
- Don't Trust Services
- Separation of Duties
- Security by Obscurity
- Keep Security Simple
- Fix Security Issues Correctly
- Software Design Patterns
- Security Patterns
- Modular Design
- Benefits of Modular Design
- The Balance Between Defense in Depth and Simplicity
- Guidelines for Avoiding Common Design Mistakes
- Avoiding Common Security Design Flaws

Topic B: Design Software to Counter Specific Threats

- The Risk Equation
- Threat Modeling
- Benefits of Threat Modeling
- Step 1: Define General Security Objectives and Scope
- Tooling and Documentation
- Assets
- Step 2: Decompose the Software
- Trust Levels
- Entry and Exit Points
- External Dependencies
- Data Flow Diagrams
- Diagramming Symbols
- Diagramming the Catalog Application
- Step 3: Identify and Rank Threats
- STRIDE
- PASTA
- Misuse Cases
- Security Zones
- Strategies for Ranking Threats
- DREAD
- Risk Response Strategies
- Severity
- Risks Outside Your Control

- Guidelines for Identifying and Ranking Threats
- Step 4: Counter Each Threat
- Countermeasures
- Identifying Threats and Countermeasures

Lesson 4: Developing Secure Code

Topic A: Follow Best Practices for Secure Coding
- Development Documentation and Deliverables
- Application and Data Integrity
- Common General Programming Errors
- Insecure Deserialization
- Guidelines for Secure Coding
- Researching Your Secure Coding Checklist
- Buffer Overrun Defects
- Buffer Overflows
- Guidelines to Prevent Buffer Overflow Defects
- Buffer Overreads
- Guidelines to Prevent Buffer Overread Defects
- Integer Overflows
- Guidelines to Prevent Integer Overflow Defects
- Uncontrolled Format Strings
- Insecure Output Encoding
- XXE Attacks
- Guidelines to Prevent Uncontrolled Format String Defects
- Race Condition
- Impact of Race Conditions on Threading/Multiprocessing
- Guidelines to Prevent Race Condition Defects
- Performing a Memory-Based Attack

Topic B: Prevent Platform Vulnerabilities
- OWASP Top Ten Platform Vulnerabilities
- Authentication
- Authorization
- Broken Authentication
- Guidelines to Prevent Web Vulnerability Defects
- Guidelines to Prevent Mobile App Vulnerability Defects
- Guidelines to Prevent Internet of Things Vulnerability Defects
- Desktop Application Vulnerabilities
- DLL Injection
- Shellcode Injection
- Debugger Security
- Differences Among Desktop Platforms
- Managed vs. Unmanaged
- Desktop Application Attack Vectors
- Development Tool and Project Configuration
- Guidelines to Prevent Desktop Application Vulnerabilities
- Finding Common Web Vulnerabilities

Topic C: Prevent Privacy Vulnerabilities
- Privacy Vulnerability Defects
- Privacy by Design
- Data Anonymization
- Guidelines to Prevent Privacy Vulnerability Defects
- Handling Privacy Defects

Lesson 5: Implementing Common Protections
Topic A: Limit Access Using Login and User Roles
- Web Sessions
- Secure Session Management
- Methods for Passing Session IDs
- Access Control
- Guidelines for Secure Session Management
- User Provisioning
- Password Recovery
- Account Lockouts
- Guidelines for Secure Password Management
- Handling Authentication and Authorization Defects
Topic B: Protect Data in Transit and At Rest
- Encryption
- Uses for Encryption
- Cryptographic Lifecycle
- Symmetric Encryption
- Asymmetric Encryption
- Hashing
- Digital Signatures
- Digital Signature Non-repudiation
- Digital Certificates
- PKI
- PKI Components
- The PKI Process
- Key Management
- Key Management Factors
- Certificate Revocation
- Guidelines for Protecting Data in Transit and at Rest
- Protecting Data in Transit and at Rest
Topic C: Implement Error Handling and Logging
- Error Handling
- Uses for Error Handling
- Error Messaging
- Logging
- Guidelines for Implementing Error Handling and Logging
- Reviewing Error Handling
- Improving Error Handling

Topic D: Protect Sensitive Data and Functions
- Sensitive Data
- Output Restrictions
- Function Level Access Control
- Case Study: Cross-Site Scripting Defect
- Guidelines for Protecting Sensitive Data and Functions
- Protecting Sensitive Data and Functions
- Staging a Persisted XSS Attack on an Administrator Function
Topic E: Protect Database Access
- Case Study: SQL Injection Defect
- Query Parameterization
- Database Connection Credential Protection
- Guidelines for Protecting Database Access
- Protecting Database Access

Lesson 6: Testing Software Security
Topic A: Perform Security Testing
- The Role of Testing
- Phases of Software Testing
- Development Testing
- Unit Testing
- Integration Testing
- Documentation and Deliverables for Testing
- Manual Inspection and Code Review
- Code Review Strategies
- Guidelines for Security Testing
- Performing Manual Inspection and Review
Topic B: Analyze Code to find Security Problems
- Static Code Analysis
- Strategies for Using Static Analysis
- Dynamic Code Analysis
- Guidelines for Code Analysis
- Performing Code Analysis
Topic C: Use Automated Testing Tools to Find Security Problems
- Automated Testing
- Unit Testing
- Guidelines for Using Automated Testing Tools
- Using a Test Suite to Automate Unit Testing

Lesson 7: Maintaining Security in Deployed Software
Topic A: Monitor and Log Applications to Support Security
- Emerging Security Problems
- Situational Awareness
- Security Monitoring
- Intrusion Detection and Prevention
- Monitor Placement
- Logging
- Guidelines for Monitoring and Logging a Deployed Application
- Monitoring and Logging a Deployed Application

Topic B: Maintain Security after Deployment
- Maintenance
- Patches and Updates
- Uninstallation and Deprovisioning
- Guidelines for Maintaining Security of Deployed Software
- Maintaining Security After Deployment

Appendix A: Mapping Course Content to Cyber Secure Coder (Exam CSC-210)