

2010 SECURITY SERIES FOR PROGRAMMING, DATABASES, & SYSTEMS

GENERAL SECURITY SEMINARS

- Understanding Web Application Security (A Technical Overview)

.NET SECURITY CLASSES

- Secure .Net Coding for Frameworks 3.5
- Securing .Net Web Applications using Frameworks 3.5
- Securing .Net Web Services using Frameworks 3.5
- Securing .Net Code, Web Applications, and Web Services for Frameworks 3.5
(Formerly know as: *.Net Security Programming or Securing .Net Applications*, now a major upgrade with more robust content, greater depth, and for Frameworks 3.5)

JAVA SECURITY CLASSES

- Secure Java-EE Development
(Formerly know as: *Securing Java Applications or Secure Java Programming*, now just upgraded with more robust content, depth, and for the latest Java Framework)
- Websphere Ver. 6 **Security** Administration and Programming

SQL SERVER SECURITY CLASSES

- SQL Server 2008 Security Essentials
- Securing SQL Server 2008 for DBA's and Best Practices
- SQL Server 2008 Security Best Practices for Developers

WINDOWS SERVER 2008 SECURITY CLASSES

- Windows Server 2008 Security Essentials
- Securing Windows Server 2008 for Administrators and Best Practices

Learning Objectives

- Understand the concepts and terminology behind defensive, secure, coding
- Appreciate the magnitude of the problems associated with web application security and the potential risks associated with those problems
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Understand the vulnerabilities of associated with authentication and authorization
- Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure
- Relate to the potential vulnerabilities and defenses for the processing of XML in web services and Ajax

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is designed for System Administrators and programmers who need to configure security and at both application level (development) and application server level (runtime).

Course Description

Understanding Web Application Security is an essential application security training course for technical leads, project managers, testing/QA personnel and other enterprise stakeholders who need to understand the issues and concepts associated with secure web applications. During this one day dynamic

seminar, students learn the best practices for designing, implementing, and deploying secure web applications. Perhaps just as significantly, students learn about current, real examples that illustrate the potential consequences of not following these best practices.

Students who attend **Understanding Web Application Security** will leave this course armed with an understanding of software vulnerabilities, defenses for those vulnerabilities, and testing those defenses for sufficiency. This course quickly introduces the most common security vulnerabilities faced by web applications today. Each vulnerability is examined through a process of describing the threat and attack mechanisms, the associated vulnerabilities, and, finally, designing, implementing, and testing effective defenses. In many cases, there are demonstrations that reinforce these concepts with real vulnerabilities, attacks, and defenses.

Prerequisites

This is course designed for web application project stakeholders who wish to get up and running on developing well defended web applications. Attendees should have a minimum of 2 years working knowledge in the

IT industry, and ideally, students should have a basic understanding of web applications and the associated technologies. Actual development working knowledge is helpful but not necessary.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Understand the concepts and terminology behind defensive, secure, coding
- Appreciate the magnitude of the problems associated with web application security and the potential risks associated with those problems
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Understand the vulnerabilities of associated with authentication and authorization
- Understand techniques and measures that can be used to harden web and application servers as well as other components in your infrastructure
- Relate to the potential vulnerabilities and defenses for the processing of XML in web services and Ajax

TOPICS COVERED IN LECTURE & LAB

Session 1: Foundation

- Misconceptions
 - Thriving Industry of Identity Theft
 - Dishonor Roll of Data Breaches
 - TJX: Anatomy of a Disaster
 - Heartland: What? Again?
- Security Concepts
 - Terminology and Players
 - Assets, Threats, and Attacks
 - OWASP
 - CWE/SANS Top 25 Programming Errors
 - Categories
 - What they mean to for web applications
- Defensive Coding Principles
 - Common Vulnerabilities and Exposures
 - OWASP Top Ten for 2010
 - Security Is a Lifecycle Issue

- Minimize Attack Surface Area
- Defense in Depth
- Manage Resources
- Layers of Defense: Tenacious D
- Compartmentalize
- Consider All Application States
- Do NOT Trust the Untrusted
- Fix Security Defects Correctly
- Leverage Experience
- Reality
 - Recent, Relevant Incidents
 - Find Security Defects In Web Application

Session 2: Top Security Vulnerabilities

- Unvalidated Input
- Broken Access Control
- Broken Authentication and Session Management
- Cross Site Scripting (XSS/CSRF) Flaws
- Injection Flaws
- Error Handling and Information Leakage
- Insecure Storage
- Insecure Management of Configuration
- Direct Object Access
- Spoofing and Redirects

Session 3: Defending XML Processing

- Defending XML
- Defending Web Services
- Defending Ajax

Session 4: Secure Software Development (SSD)

- SSD Process Overview
- Applying Processes and Practices
- Risk Analysis

Session 5: Security Testing

- Testing Principles
- Reviews as Form of Testing
- Testing
- Tools
- Testing Practices

Learning Objectives

2010 SECURITY SERIES

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Learn the entire spectrum of threats and attacks that take place against software applications in today's world
- Use Threat Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net applications
- Understand the vulnerabilities of the .Net programming language and the runtime environment as well as how to harden both
- Understand and work with .Net platform security to gain an appreciation for what is protected and how
- Understand the basics of Cryptography and Encryption and where they fit in the overall security picture
- Work with the .Net Cryptographic services
- Examine how role-based security works in .Net and use it to control access
- Examine how Code Access Security (CAS) works and use it to control access
- Understand and work with the mechanics of isolated storage
- Understand the fundamentals of XML Digital Signature and XML Encryption

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for students whom have Intermediate to Advanced experience with .Net Frameworks 3.0 or higher.

Course Description

Secure .Net Coding is a hands-on, lab-intensive .Net security, code-level training course that teaches students the best practices for designing, implementing, and deploying secure programs in .Net. Students will take an application from requirements through to implementation, analyzing and testing for software vulnerabilities. This course explores well beyond basic programming skills, teaching developers sound processes and practices to apply to the entire software development lifecycle. Perhaps just as significantly, students learn

about current, real examples that illustrate the potential consequences of not following these best practices.

This course is short on theory and long on application, providing students with in-depth, code-level labs. Students who attend Secure .Net Coding will leave the course armed with the required skills to recognize software vulnerabilities (actual and potential) and implement defenses for those vulnerabilities. This course quickly introduces developers to the various types of threats against their software. The concept and process of Threat Modeling is introduced as a key enabler for implementing effective and appropriate security for software and information assets. This course includes coverage of the many security-related technologies and APIs that exist in the .Net world.

This class is “technology-centric”, designed to train attendees in essential secure coding and development skills, coupling the most current, effective techniques with the soundest industry practices. This workshop is about **50% dynamic lab exercises** and **50% lecture**. The course provides a solid foundation in basic terminology and concepts, extended and built upon throughout the engagement.

Students will examine various recognized attacks against web applications. Processes and best practices are discussed and illustrated through both discussions and group activities.

The second portion of the course steps through a series of vulnerabilities illustrating in very real terms the right way to implement secure .Net applications. The last portion of the course examines several design patterns that can be used to facilitate better application architecture, design, implementation, and deployment.

Prerequisites

This is an intermediate level .Net programming course designed for application project stakeholders who wish to get up and running on developing well defended .Net applications. Familiarity with the C# programming language is required, and real world programming experience is highly recommended.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Learn the entire spectrum of threats and attacks that take place against software applications in today's world
- Use Threat Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net applications
- Understand the vulnerabilities of the .Net programming language and the runtime environment as well as how to harden both
- Understand and work with .Net platform security to gain an appreciation for what is protected and how
- Understand the basics of Cryptography and Encryption and where they fit in the overall security picture
- Work with the .Net Cryptographic services
- Examine how role-based security works in .Net and use it to control access
- Examine how Code Access Security (CAS) works and use it to control access
- Understand and work with the mechanics of isolated storage
- Understand the fundamentals of XML Digital Signature and XML Encryption

TOPICS COVERED IN LECTURE & LAB

Session: Defensive Coding Overview

- Misconceptions
 - Thriving Industry of Identify Theft
 - Dishonor Roll of Data Breaches
 - TJX: Anatomy of a Disaster
 - Heartland: What? Again?
- Security Concepts
 - Terminology and Players
 - Assets, Threats, and Attacks
 - OWASP
 - CWE/SANS Top 25 Programming Errors
 - Categories
 - What they mean to your applications
- Defensive Coding Principles
 - Security Is a Lifecycle Issue
 - Bolted on Versus Baked
 - Minimize Attack Surface Area
 - Examples of Minimization
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Learning From Vulnerabilities
- Reality
 - Recent, Relevant Incidents
 - Finding Security Defects In Web Application

Session: Vulnerabilities

- Unvalidated Input – XSS/CSRF, Injection, and Others
- Broken Authentication and Authorization
- Information Leakage - Error Handling, Logging, Insecure Storage and Others
- Spoofing - Protecting Your Users and Your Applications

Session: .Net Security Fundamentals

- .Net Security Overview
- Services Provided
- Code Protections
- Data Protections

.NET Assembly Security

- The role of Application Domains
- Protecting assemblies from tampering
- Using obfuscation
- Using publisher certificates
- Using FxCop.exe

Session: Cryptography Overview

- Cryptography defined
- Strong Encryption
- Ciphers and algorithms
- Message digests
- Keys and key management
- Types of keys
- Key management
- Certificate management
- Encryption/Decryption

.NET Cryptographic Services

- The role of cryptographic services
- Hash algorithms and hash codes
- Generating hashed data
- Validating hash codes
- Encryption and decryption
- Encrypting data symmetrically
- Encrypting data asymmetrically

Understanding Role Based Security

- Using role based security
- Creating and administering roles
- Principals, identity and roles
- Determining role membership
- Restricting actions based on roles

Code Access Security

- What is Code Access Security (CAS)
- CAS components
- Using CAS to secure applications”
- Interacting with CAS

Isolated Storage

- The purpose of Isolated Storage
- Levels of isolated storage
- Using isolated storage administrative tools
- Working with isolated storage programmatically

Session: Defending XML Processing

- Defending XML
 - Understanding common attacks and how to defend
 - Operating in safe mode
 - Using standards-based security
 - XML-aware security infrastructure

Session: Understanding What's Important

- **Prioritizing Your Efforts**
 - Common Vulnerabilities and Exposures
 - OWASP Top Ten for 2010
 - Security Is a Lifecycle Issue
 - Minimize Attack Surface Area
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Leverage Experience

Learning Objectives

2010 SECURITY SERIES

- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, *cross-site scripting*, and *injections*
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Prevent and defend the many potential vulnerabilities associated with untrusted data
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind *Cross-Site Scripting (XSS)* and *Injection attacks*
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Be able to detect, attack, and implement defenses against *XSS* and *Injection attacks*
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in web applications
- Design and develop strong, robust authentication and authorization implementations within the context of .NET
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Be able to detect, attack, and implement defenses for XML-based services and functionality
- Understand techniques and measures that can be used to harden web and application servers as well as other components in your infrastructure

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for students who have been, or intend to design Web-Based Applications using Microsoft's .Net Frameworks 3.0 or higher.

Course Description

Students who attend **Securing .Net Web Applications** will leave the course armed with the skills required to recognize actual and potential software vulnerabilities, implement defenses for those vulnerabilities, and test those defenses for sufficiency.

This course quickly introduces developers to the most common security vulnerabilities faced by web applications today. Each vulnerability is examined from a Net perspective through a process of describing the threat and attack mechanisms, recognizing associated vulnerabilities, and, finally, designing, implementing, and testing effective defenses. In many cases, there are labs that reinforce these concepts with real vulnerabilities and attacks. Students are then challenged to design and implement the layered defenses they will need in defending their own applications.

This class is "technology-centric", designed to train attendees in essential secure coding and development skills, coupling the most current, effective techniques with the soundest industry practices. This workshop is about **50% dynamic lab exercises** and **50% lecture**.

The course provides a solid foundation in basic terminology and concepts, extended and built upon throughout the engagement. Students will examine various recognized attacks against web applications. Processes and best practices are discussed and illustrated through both discussions and group activities.

The second portion of the course steps through a series of vulnerabilities illustrating in very real terms the right way to implement secure .Net applications. The last portion of the course examines several design patterns that can be used to facilitate better application architecture, design, implementation, and deployment.

Prerequisites

This is an **intermediate -level** .Net programming course, designed for developers who wish to get up and running on developing well defended software applications. This course may be customized to suit your team's unique objectives.

Familiarity with C# is required and real world programming experience is highly recommended. Ideally students should have approximately (6) months to a year of .Net application development and practical experience.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, *cross-site scripting*, and *injections*
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Prevent and defend the many potential vulnerabilities associated with untrusted

- data
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind *Cross-Site Scripting (XSS)* and *Injection attacks*
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Be able to detect, attack, and implement defenses against *XSS* and *Injection attacks*
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in web applications
- Design and develop strong, robust authentication and authorization implementations within the context of .NET
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Be able to detect, attack, and implement defenses for XML-based services and functionality
- Understand techniques and measures that can be used to harden web and application servers as well as other components in your infrastructure

TOPICS COVERED IN LECTURE & LAB

Session: Foundation

- Misconceptions
 - Thriving Industry of Identify Theft
 - Dishonor Roll of Data Breaches
 - TJX: Anatomy of a Disaster
 - Heartland: What? Again?
- Security Concepts
 - Terminology and Players
 - Assets, Threats, and Attacks
- OWASP
 - CWE/SANS Top 25 Programming Errors
 - Categories
 - What they mean to your applications
- Defensive Coding Principles
 - Security Is a Lifecycle Issue
 - Bolted on Versus Baked
 - Minimize Attack Surface Area
 - Examples of Minimization
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Learning From Vulnerabilities
- Reality
 - Recent, Relevant Incidents
 - Finding Security Defects In Web Applications

Session: Top Security Vulnerabilities

- Unvalidated Input
 - Unvalidated Input: Description
 - Integer Arithmetic Vulnerabilities
 - Unvalidated Input: From the Web
 - Hidden Values in HTTP Communications
 - Unvalidated Input: Symptoms and Detection
 - Detection Through Fuzz Testing
 - Unvalidated Input: Fixes
 - Identifying Trust Boundaries
 - Designing An Appropriate Response
 - Testing Defenses And Responses
- Overview of Regular Expressions
 - Description with working example
- Broken Access Control
 - Access Control Issues
 - Broken Access Control: Description
 - Excessive Privileges
 - Unprotected URL/Resource Access: Description
 - Unprotected URL/Resource Access: Symptoms and Detection
 - Primary Concerns in URL/Resource Access
 - Unprotected URL/Resource Access: Fixes
 - Protecting Sessions
 - Addressing Client-Side Caching of Content
 - .Net authorization security overview
 - Defending special privileges such as administrative functions
 - Application authorization best practices
- Broken Authentication and Session Management
 - Description with working example
 - Defenses

- Multi-layered defenses of authentication services
- Password management strategies
- Password handling with hashing
- Mitigating password caching
- Testing defenses and responses for weaknesses
- Alternative authentication mechanisms
- Best practices for session management
- Defending session hijacking attacks
- Best practices for Single Sign-On (SSO)
- Description With Working Example
- Defenses
- Character Encoding Complications
- Blacklisting
- Whitelisting
- HTML/XML Entity Encoding
- Trust Boundary Definition
- Implementing An Effective Layered Defense
- Designing An Appropriate Response
- Injection Flaws
 - SQL Injection Continues to be Prevalent
 - Injection Flaws: Description
 - Injection Flaws: Symptoms and Detection
 - SQL Injection Examples
 - SQL Injection Attacks Evolve
 - Attackers have a Variety of Tools
 - SQL Injection: Drill Down on Stored Procedures
 - SQL Injection: Drill Down on ORM
 - Minimize SQL Injection Vulnerabilities
 - Minimizing Injection Flaws
 - Command Injection Vulnerabilities
- Error Handling and Information Leakage
 - Description with working example
 - Defenses
 - .Net web application exception handling framework
 - Error response best practices
 - Error, auditing, and logging content management
 - Error, auditing, and logging service management
 - Best practices for supporting web attack forensics
- Insecure Storage
 - Description with working example
 - Defenses
 - Data leakage
 - Risk minimization
 - Cryptography Overview
 - Data encryption
 - Partial/Complete
 - Property/Deployment/Configuration files
 - In-Memory Data Handling
 - Handling Passwords on Server Side
- Insecure Management of Configuration
 - Description with working example

- Defenses
- System hardening
- .Net application server configuration "Gotchas!"
- Hardening software installation
- Direct Object Access
 - Description with working example
 - Defenses
 - XML/DTD/Schema/XSLT best practices
 - Race Conditions
 - Direct Object References
- Spoofing and Redirects
 - Spoofing: Description
 - Name Resolution Vulnerabilities
 - Attacks are Constant and Changing
 - Spoofing: Fixes
 - Cross Site Request Forgeries (CSRF)
 - How To Get Victim To Select URL?
 - CSRF Defenses are Entirely Server-Side
 - CSRF Defenses are Evolving
 - Redirects and Forwards
 - Safe Redirects and Forwards

Session: Understanding What's Important

- Prioritizing Your Efforts
 - Common Vulnerabilities and Exposures
 - OWASP Top Ten for 2010
 - Security Is a Lifecycle Issue
 - Minimize Attack Surface Area
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Leverage Experience

Session: Defending XML Processing

- Defending XML
 - Understanding common attacks and how to defend
 - Operating in safe mode
 - Using standards-based security
 - XML-aware security infrastructure
- Defending Web Services
 - Security exposures
 - Transport-level security
 - Message-level security
 - WS-Security
 - Attacks and defenses
- Defending Ajax
 - Ajax Security exposures
 - Attack surface changes
 - Injection threats and concerns
 - Effective defenses and practices

Learning Objectives

2010 SECURITY SERIES

- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Prevent and defend the many potential vulnerabilities associated with untrusted data
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Be able to detect, attack, and implement defenses against XSS and Injection attacks
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net-based web applications
- Understand the basics of XML Encryption as well as how it can be used as part of the defensive infrastructure for web services
- Understand the basics of XML Digital Signature as well as how it can be used as part of the defensive infrastructure for web services
- Understand and defend vulnerabilities that are specific to XML and XML parsers

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the

intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for Web Developers who are using Microsoft's .Net Frameworks 3.0 or higher and have experience building Web Applications.

Course Description

Securing .Net Web Services is a lab-intensive, hands-on .Net security training course, essential for experienced enterprise developers who need to produce secure .Net-based web services. In addition to teaching basic programming skills, this course digs deep into sound processes and practices that apply to the entire software development lifecycle.

Designing, implementing, and deploying secure services presents unique challenges. In addition to dealing with all of the vulnerabilities and attacks associated with web applications, web services must address business-oriented concerns such as authentication, authorization, non-repudiation and others. The complicating factor is that all measures must be implemented within the constraints of standards and high-level s of interoperability.

In this course, students thoroughly examine best practices for defensively coding .Net services, including XML processing. Students will repeatedly attack and then defend various assets associated with fully-functional web services. This hands-on approach drives home the mechanics of how to secure .Net web services in the most practical of terms.

Security experts agree that the least effective approach to security is "penetrate and patch". It is far more effective to "bake" security into an application throughout its lifecycle. After spending significant time trying to defend a poorly designed (from a security perspective) web application, developers are ready to learn how to build secure web applications

starting at project inception. The final portion of this course builds on the previously learned mechanics for building defenses by exploring how design and analysis can be used to build stronger applications from the beginning of the software lifecycle.

Prerequisites

This is an **intermediate-level** .Net programming course designed for application project stakeholders who wish to get up and running on developing well defended .Net applications. Familiarity with the C# programming language is required, and real world programming experience is highly recommended.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Prevent and defend the many potential vulnerabilities associated with untrusted data
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Be able to detect, attack, and implement defenses against XSS and Injection attacks
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against

meaningful assets

- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net-based web applications
- Understand the basics of XML Encryption as well as how it can be used as part of the defensive infrastructure for web services
- Understand the basics of XML Digital Signature as well as how it can be used as part of the defensive infrastructure for web services
- Understand and defend vulnerabilities that are specific to XML and XML parsers

TOPICS COVERED IN LECTURE & LAB

Session: Foundation

- Misconceptions
 - Thriving Industry of Identify Theft
 - Dishonor Roll of Data Breaches
 - TJX: Anatomy of a Disaster
 - Heartland: What? Again?
- Security Concepts
 - Terminology and Players
 - Assets, Threats, and Attacks
 - OWASP
 - CWE/SANS Top 25 Programming Errors
 - Categories
 - What they mean to your services
- Defensive Coding Principles
- Reality
 - Recent, Relevant Incidents
 - Find Security Defects In Web Application

Session: SOA Security Overview

- Challenges
 - Identity and Propagation
 - Real-time Transactions
 - Diverse Environments
 - Information Protection
 - Standards compliance
- Services and Security
 - SOA Components
 - Service Lifecycle
 - Security Policies
- Security Services
 - Identity
 - Authentication
 - Authorization
 - Confidentiality/Integrity
 - Auditing
 - Non-repudiation

Session: Applying Security to Services

- Direct Service Exposure
- Indirect Service Exposure
- Enterprise Service Bus (ESB)
 - Mediating Security Services
 - Transport-Level Security
 - Message-Level Security
 - Policy Enforcement
 - Policy Management
 - Protecting the ESB
- Composed Services
 - Single-Sign On
 - Trust Relationships
 - Trust Relationships and Web Services

Session: Defending XML Processing

- Defending XML
 - Understanding common attacks and how to defend
 - Operating in safe mode
 - Using standards-based security
 - XML-aware security infrastructure
- Defending Web Services
 - Security exposures
 - Transport-level security
 - Message-level security
 - WS-Security
 - Attacks and defenses

Session: WS-Security

- WS-Security
 - WS-Security Stack
 - Best Practices
- XML Digital Signature
 - Architecture
 - Working with XML Digital Signature
 - Integrating XML Digital Signature into Web Services
 - Best Practices

Session: Top Security Vulnerabilities

- Unvalidated Input
- Overview of Regular Expressions
- Broken Access Control
- Broken Authentication and Session Management
- Cross Site Scripting (XSS/CSRF) Flaws
- Injection Flaws
- Error Handling and Information Leakage
- Insecure Storage
- Insecure Management of Configuration
- Direct Object Access

- Spoofing

Session: Best Practices

- Best Practices and Principles
 - Security Is A Lifecycle Issue
 - Minimize Attack Surface
 - Manage Resources
 - Application States
 - Compartmentalize
 - Defense In Depth - Layered Defense
 - Consider All Application States
 - Not Trusting The Untrusted
 - Security Defect Mitigation
 - Leverage Experience
- Web Application Security Design Patterns
 - Authentication Enforcer
 - Authorization Enforcer
 - Intercepting Validator
 - Secure Base Action
 - Secure Logger
 - Secure Pipe
 - Secure Service Proxy
 - Intercepting Web Agent

Session: Secure Software Development (SSD)

- SSD Process Overview
 - CLASP Defined
 - CLASP Applied
- Asset, Boundary, and Vulnerability Identification
- Vulnerability Response
- Design and Code Reviews
- Applying Processes and Practices
- Risk Analysis

Session: Security Testing

- Testing as Lifecycle Process
- Testing Planning and Documentation
- Testing Tools And Processes
 - Principles
 - Reviews
 - Testing
 - Tools
- Static and Dynamic Code Analysis
- Testing Practices
 - Authentication Testing
 - Session Management Testing
 - Data Validation Testing
 - Denial Of Service Testing
 - Web Services Testing

Learning Objectives

2010 SECURITY SERIES

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Learn the entire spectrum of threats and attacks that take place against software applications in today's world
- Use Threat Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net applications
- Understand the vulnerabilities of the .Net programming language and the runtime environment as well as how to harden both
- Understand the basics of Cryptography and Encryption and where they fit in the overall security picture
- Work with the .Net Cryptographic services
- Examine how role-based security works in .Net and use it to control access
- Examine how Code Access Security (CAS) works and use it to control access
- Understand and work with the mechanics of isolated storage
- Understand the fundamentals of XML Digital Signature and XML Encryption
- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Understand the vulnerabilities of associated with authentication and authorization
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Understand techniques and measures that can be used to harden web and application servers as well as other components in your infrastructure

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for Web Developers who are using Microsoft's .Net Frameworks 3.0 or higher and have a minimum of (6) months experience building Applications.

Course Description

This course is a hands-on, lab-intensive .Net security, code-level training course that teaches students the best practices for designing, implementing, and deploying secure programs in the .Net environment for either Server-Side, Web Applications, or Implementing Web Services. Students will take an application from requirements through to implementation, analyzing and testing for software vulnerabilities. This course explores well beyond basic programming skills, teaching developers sound processes and practices to apply to the entire software development lifecycle. Perhaps just as significantly, students learn about current, real examples that illustrate the potential consequences of not following these best practices. This course is short on theory and long on application, providing students with in-depth, code-level labs.

Designing, implementing, and deploying secure services presents unique challenges. In addition to dealing with all of the vulnerabilities and attacks associated with web applications, web services must address business-oriented concerns such as authentication, authorization, non-repudiation and others. The complicating factor is that all measures must be

implemented within the constraints of standards and high-level s of interoperability.

In this course, students thoroughly examine best practices for defensively coding .Net services, including XML processing. Students will repeatedly attack and then defend various assets associated with fully-functional web services. This hands-on approach drives home the mechanics of how to secure .Net web services in the most practical of terms.

Security experts agree that the least effective approach to security is "penetrate and patch". It is far more effective to "bake" security into an application throughout its lifecycle. After spending significant time trying to defend a poorly designed (from a security perspective) web application, developers are ready to learn how to build secure web applications starting at project inception. The final portion of this course builds on the previously learned mechanics for building defenses by exploring how design and analysis can be used to build stronger applications from the beginning of the software lifecycle.

Prerequisites

This is an **Intermediate-Level** .Net programming course designed for application project stakeholders who wish to get up and running on developing well defended .Net applications. Familiarity with the **C# programming language** is required, and real world programming experience is highly recommended.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets

- Learn the entire spectrum of threats and attacks that take place against software applications in today's world
- Use Threat Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net applications
- Understand the vulnerabilities of the .Net programming language and the runtime environment as well as how to harden both
- Understand the basics of Cryptography and Encryption and where they fit in the overall security picture
- Work with the .Net Cryptographic services
- Examine how role-based security works in .Net and use it to control access
- Examine how Code Access Security (CAS) works and use it to control access
- Understand and work with the mechanics of isolated storage
- Understand the fundamentals of XML Digital Signature and XML Encryption
- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Understand the vulnerabilities of associated with authentication and authorization
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Understand techniques and measures that can be used to harden web and application servers as well as other components in your infrastructure

TOPICS COVERED IN LECTURE & LAB

Session: Defensive Coding Overview

- Misconceptions
 - Thriving Industry of Identity Theft
 - Dishonor Roll of Data Breaches
 - TJX: Anatomy of a Disaster
 - Heartland: What? Again?
- Security Concepts
 - Terminology and Players
 - Assets, Threats, and Attacks
 - OWASP
 - CWE/SANS Top 25 Programming Errors
 - Categories
 - What they mean to your applications
- Defensive Coding Principles
 - Security Is a Lifecycle Issue
 - Bolted on Versus Baked
 - Minimize Attack Surface Area
 - Examples of Minimization
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Learning From Vulnerabilities
- Reality
 - Recent, Relevant Incidents
 - Finding Security Defects In Web Application

Session: Vulnerabilities

- Unvalidated Input
 - Unvalidated Input: From the Web
 - Hidden Values in HTTP Communications
 - Fuzz Testing
 - Testing Defenses And Responses
- Overview of Regular Expressions
- Broken Access Control
 - Access Control Issues
 - Unprotected URL/Resource Access: Symptoms and Detection
 - Protecting Sessions
 - Addressing Client-Side Caching of Content
 - Application authorization best practices
- Broken Authentication and Session Management
 - Defenses
 - Multi-layered defenses of authentication services
 - Password management strategies
 - Testing defenses and responses for

- weaknesses
- Defending session hijacking attacks
- Single Sign-On (SSO)
- Blacklisting/Whitelisting
- HTML/XML Entity Encoding
- Effective Layered Defense
- Injection Flaws
 - SQL Injection Examples
 - SQL Injection Attacks Evolve
 - Minimizing Injection Flaws
- Error Handling and Information Leakage
 - .Net web application exception handling framework
 - Best practices for supporting web attack forensics
- Insecure Storage
 - Data leakage
 - Cryptography Overview
 - Data encryption
 - Handling Passwords on Server Side
- Insecure Management of Configuration
 - System hardening
 - .Net application server configuration "Gotchas!"
 - Hardening software installation
- Direct Object Access
 - XML/DTD/Schema/XSLT best practices
 - Race Conditions
 - Direct Object References
- Spoofing and Redirects
 - Spoofing: Description
 - Name Resolution Vulnerabilities
 - Cross Site Request Forgeries (CSRF)
 - Safe Redirects and Forwards

Session: .Net Security Fundamentals

- .Net Security Overview
- Services Provided
- Code Protections
- Data Protections
- NET Assembly Security
- The role of Application Domains
- Protecting assemblies from tampering
- Using obfuscation
- Using publisher certificates
- Using FxCop.exe

Session: Cryptography Overview

- Cryptography defined
- Strong Encryption
- Ciphers and algorithms
- Message digests
- Keys and key management

- Types of keys
- Key management
- Certificate management
- Encryption/Decryption

.NET Cryptographic Services

- The role of cryptographic services
- Hash algorithms and hash codes
- Generating hashed data
- Validating hash codes
- Encryption and decryption
- Encrypting data symmetrically
- Encrypting data asymmetrically

Understanding Role Based Security

- Using role based security
- Creating and administering roles
- Principals, identity and roles
- Determining role membership
- Restricting actions based on roles

Code Access Security

- What is Code Access Security (CAS)
- CAS components
- Using CAS to secure applications"
- Interacting with CAS

Isolated Storage

- The purpose of Isolated Storage
- Levels of isolated storage
- Using isolated storage administrative tools
- Working with isolated storage programmatically

Session: Defending XML Processing

- Defending XML
 - Understanding common attacks and how to defend
 - Operating in safe mode
 - Using standards-based security
 - XML-aware security infrastructure
- Defending Ajax
 - Ajax Security exposures
 - Attack surface changes
 - Injection threats and concerns
 - Effective defenses and practices

Session: SOA Security Overview

- Challenges
 - Identity and Propagation
 - Real-time Transactions
 - Diverse Environments
 - Information Protection
 - Standards compliance
 - Services and Security
 - SOA Components
 - Service Lifecycle
 - Security Policies
- Security Services
 - Identity
 - Authentication
 - Authorization
 - Confidentiality/Integrity
 - Auditing
 - Non-repudiation

Session: Applying Security to Services

- Direct Service Exposure
- Indirect Service Exposure
- Enterprise Service Bus (ESB)
 - Mediating Security Services
 - Transport-Level Security
 - Message-Level Security
 - Policy Enforcement
 - Policy Management
 - Protecting the ESB
- Composed Services
 - Single-Sign On
 - Trust Relationships
 - Trust Relationships and Web Services

Session: WS-Security

- WS-Security
 - WS-Security Stack
 - Best Practices
- XML Digital Signature
 - Architecture
 - Working with XML Digital Signature
 - Integrating XML Digital Signature into Web Services
 - Best Practices

Session: Best Practices

- Best Practices and Principles
 - Security Is A Lifecycle Issue
 - Minimize Attack Surface
 - Manage Resources
 - Application States
 - Compartmentalize
 - Defense In Depth - Layered Defense
 - Consider All Application States
 - Not Trusting The Untrusted

- Security Defect Mitigation
- Leverage Experience
- Web Application Security Design Patterns
 - Authentication Enforcer
 - Authorization Enforcer
 - Intercepting Validator
 - Secure Base Action
 - Secure Logger
 - Secure Pipe
 - Secure Service Proxy
 - Intercepting Web Agent

Session: Secure Software Development (SSD)

- SSD Process Overview
 - CLASP Defined
 - CLASP Applied
- Asset, Boundary, and Vulnerability Identification
- Vulnerability Response
- Design and Code Reviews
- Applying Processes and Practices
- Risk Analysis

Session: Security Testing

- Testing as Lifecycle Process
- Testing Planning and Documentation
- Testing Tools And Processes
 - Principles
 - Reviews
 - Testing
 - Tools
- Static and Dynamic Code Analysis
- Testing Practices
 - Authentication Testing
 - Session Management Testing
 - Data Validation Testing
 - Denial Of Service Testing
 - Web Services Testing

Session: Understanding What's Important

- Prioritizing Your Efforts
 - Common Vulnerabilities and Exposures
 - OWASP Top Ten for 2010
 - Security Is a Lifecycle Issue
 - Minimize Attack Surface Area
 - Defense in Depth
 - Manage Resources
 - Layers of Defense: Tenacious D
 - Compartmentalize
 - Consider All Application States
 - Do NOT Trust the Untrusted
 - Fix Security Defects Correctly
 - Leverage Experience

Learning Objectives

2010 SECURITY SERIES

- Be prepared to develop secure Java web applications, or to secure existing applications by refactoring as necessary
- Instruct the web container to enforce your desired authentication and authorization policies
- Validate user input aggressively, for general application health and to foil specific attacks including injection and XSS
- Be on guard against common web-application attack vectors such as cross-site scripting, injection, and request forgery, and take appropriate counter-measures
- Configure a server and/or application to use one-way or two-way HTTPS
- Use cryptography appropriately in application design and implementation, including signature and signature verification, encryption, hashing, and tactical use of secure random numbers

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for experienced Java developers who want to be able to follow secure development practice and to secure new and existing standalone, web, and enterprise applications.

Course Description

This advanced course shows experienced developers of Java enterprise applications how to secure those applications and to apply best practices with regard to secure coding. Authentication, authorization, and input validation are major themes, and participants use cryptographic algorithms

(via the JCA and JCE) for specific development scenarios. The course also includes thorough discussions and hands-on exercises in common web-application concerns and hacks (XSS, injection, etc.), HTTPS configuration and certificate management, error handling, logging, and auditing.

Prerequisites

- Java programming experience is essential, including understanding of OO practices, exception handling and multi-threading.
- Servlets programming experience is recommended but not required.
- JSP page-authoring experience is recommended but not required.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Be prepared to develop secure Java web applications, or to secure existing applications by refactoring as necessary

- Instruct the web container to enforce your desired authentication and authorization policies
- Validate user input aggressively, for general application health and to foil specific attacks including injection and XSS
- Be on guard against common web-application attack vectors such as cross-site scripting, injection, and request forgery, and take appropriate counter-measures
- Configure a server and/or application to use one-way or two-way HTTPS
- Use cryptography appropriately in application design and implementation, including signature and signature verification, encryption, hashing, and tactical use of secure random numbers

TOPICS COVERED IN LECTURE & LAB

Module 1: Java SE Security

- The AccessController Class Utility
- Permissions, Code Sources, and Protection Domains
- The Policy Class
- Policy Files

Module 2: Secure Coding Practice

- Code Injection
- Final Classes and Methods
- Inner Classes
- Singletons, Factories, and Flyweights
- Methods, Collections, and Data Hiding
- Object Serialization
- Software Bugs as Security Holes
- Overflows, Overruns, and Wraparounds
- Race Conditions

Module 3: Java EE Security

- Threats and Attack Vectors
- Server, Network, and Browser Vulnerabilities
- Secure Design Principles
- GET vs. POST
- Container Authentication and Authorization
- HTML Forms
- Privacy Under /WEB-INF
- HTTP and HTTPS
- Other Cryptographic Practices

- SOA and Web Services
- The OWASP Top 10

Module 4: Authentication and Authorization

- HTTP BASIC and DIGEST Authentication Schemes
- Declaring Security Constraints
- User Accounts
- Safeguarding Credentials in Transit
- Replay Attacks
- Authorization Over URL Patterns
- Roles
- FORM Authentication
- Login Form Design
- EJB Authorization
- Programmatic Security
- Programmatic Security in JSF

Module 5: Special Concerns for Web Applications

- Single Points of Decision
- Cross-Site Scripting
- Validation vs. Output Escaping
- Forceful Browsing
- Cross-Site Request Forgery
- Request Tokens
- Injection Attacks
- Protections in JDBC and JPA
- Session Management
- Taking Care of Cookies
- Validating User Input
- Validation Practices
- Whitelist Validation
- Blacklist Validation
- Regular Expressions
- Constructing REs
- Using REs in Java
- JSF Validation

Module 6: HTTPS and Certificates

- Digital Cryptography
- Encryption
- SSL and Secure Key Exchange
- Hashing
- Signature
- Keystores
- keytool
- Why Keys Aren't Enough
- X.509 Certificates

- Certificate Authorities
- Obtaining a Signed Certificate
- Configuring HTTPS
- Client-Side Certificates and Two-Way SSL
- PKCS #12 and Trust Stores
- CLIENT-CERT Authentication

Module 7: Cryptography in Java SE

- The Java Cryptography Architecture
- The KeyStore API
- The Signature Class
- The SignedObject Class
- Signing and Sealing JAR Files
- Verifying JAR Signatures
- Authorizing Code by Signature

Module 8: Cryptography in Web Applications

- Secure Random Number Generation
- The MessageDigest Class
- The Java Cryptography Extensions
- The SecretKey and KeyGenerator Types
- The Cipher Class
- Choosing Algorithms and Key Sizes
- Dangerous Practices
- Using JSSE

Module 9: Secure Development Practices

- Secure Development Cycle
- Error Handling and Information Leakage
- Failing to a Secure Mode
- Logging Practices
- Appropriate Content for Logs
- Auditing
- Strategies: Filters, Interceptors, and
- Command Chains
- Penetration Testing
- Back Doors

Learning Objectives

2010 SECURITY SERIES

- Configure global security in WebSphere Application Server
- Integrate WebSphere Application Server with LDAP
- Create and deploy a secure web application
- Configure role based security for EJBs
- Configure Data Source security and understand how Prepared Statements increases security
- Configure Single Sign-On
- Implement a custom user registry
- Understand what's involved in Web Services, messaging and J2C security
- Configure SSL in IBM HTTP Server

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is designed for System Administrators and programmers who need to configure security and at both application level (development) and application server level (runtime).

Course Description

This course delves deep into the security administration of WebSphere Application Server v6. It also teaches the security programming model of J2EE. Creating secure applications and web sites requires close cooperation between the developers and the administrators. Keeping that in mind, this course is targeted towards the developer and the administrator community.

Prerequisites

The participant should have a good understanding of Java and web technologies (Servlets, JSPs and EJBs), operational skills for Windows and basic administration skills for WebSphere application server.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Configure global security in WebSphere Application Server
- Integrate WebSphere Application Server with LDAP
- Create and deploy a secure web application
- Configure role based security for EJBs
- Configure Data Source security and understand how Prepared Statements increases security
- Configure Single Sign-On
- Implement a custom user registry
- Understand what's involved in Web Services, messaging and J2C security
- Configure SSL in IBM HTTP Server

TOPICS COVERED IN LECTURE & LAB

Common Security Threats

- Overview
- Input Data Validation
- Data Ownership Validation
- SQL Injection Problem
- SQL Injection Solution
- Malicious File Execution Problem
- Malicious File Execution Solution
- Web Authentication Mechanism
- Insecure Authentication Mechanism
- Failure to Restrict URL Access Problem
- Failure to Restrict URL Access Solution
- Cross Site Scripting (XSS) Problem
- Cross Site Scripting (XSS) Solution
- Cross Site Request Forgery (CSRF) Problem
- Cross Site Request Forgery (CSRF) Solution
- Information Leakage and Improper Error Handling Problem
- Information Leakage and Improper Error Handling Solution
- Buffer Overflow
- Buffer Overflow Example
- More Buffer Overflows
- Buffer Overflow Solution
- Insecure Communications
- Insecure Cryptographic Storage Problem
- Insecure Cryptographic Storage Solution
- Insecure Direct Object Reference
- Message Replay Attack Problem
- Message Replay Attack Solution
- Summary
- References

WebSphere Security

- Objectives
- Security Overview
- Architecture Components
- Security Components
- Digital Certificates
- SSL (Secure Sockets Layer)
- SSL in WebSphere
- Java Security
- JAAS
- CSiv2
- J2EE Security

- Authentication and Authorization
- User Registry
- Authentication Mechanism
- Global Security Configuration
- LTPA
- Single Signon (SSO)
- Configuring LTPA
- Admin Console Roles
- Stopping Secure Servers
- WebSphere Security Questions
- WebSphere Security Answers
- Reference

Configuring WebSphere Security

- Overview
- WebSphere Security
- Security Tasks
- User Registries
- WebSphere User Registries
- LDAP
- LDAP Security Basics
- LDAP Data Structure
- Example
- Distinguished Name (DN)
- DN and RDN Example
- Loading Users in Tivoli Directory Server 6.0
- Creating Users and Groups in Domino Server
- Local OS
- Custom Registry
- Precaution
- Selecting A Registry
- Configure the LDAP User Registry
- Configuring Domino Server
- Configuring Domino Server with WAS
- Configure Local OS Registry
- Enable Global Security
- Console Users
- Console Roles
- Console Role Mapping
- Make It So!
- Stopping Secure Servers
- Summary
- WebSphere Security Questions
- WebSphere Security Answers
- Resources

Securing The Installation

- Overview
- The Operating System
- Pre-Installation Tasks
- Windows Security Policy
- Unix - Umask Value
- Linux / Solaris Shadow File
- Post-Installation Tasks
- Securing Windows Files
- Securing UNIX Files
- UNIX File System
- Running Application Server as non-root User UNIX Platform
- Overview
- Review Questions
- Answers
- References

Web Application Security

- Overview
- Servlet Security
- Setting up Servlet Security
- Defining Roles
- Create a Security Constraint
- Configuring Declarative Security Using RAD
- Defining Roles Using RAD
- Defining Security Constraint Using RAD
- Configuring Declarative Security Using RAD
- Defining Roles at Application Level
- Defining Roles At Application Level Using RAD
- J2EE Role Management
- Sample Role Mapping
- Mapping Roles to Users and Groups in WebSphere
- Authentication Mechanism
- Configuring Authentication Mechanism Using RAD
- HTTP Basic Authentication
- HTTP Digest Authentication
- Form-based Authentication
- HTTPS Client Authentication
- **Lab**
- User Context of a Servlet Execution
- Accessing User Credentials
- User Context Used by RequestDispatcher
- User Context Used When Invoking an EJB
- Specifying User Context
- Specifying User Context

- Specifying User Context
- Configuring Run As Identity Using RAD
- Mapping Run As Roles to Users Using WebSphere
- The init method
- Programmatic Role-based Security
- Creating Role Sensitive Views
- Security Role References
- Configuring Security Role Reference Using RAD
- **Lab**
- Problems with Basic Authentication
- Set Up Form-based Authentication
- Create an HTML Form
- Configure a login-config Element
- Configuring a login-config Element using RAD
- Handling Login Failure
- Protecting Session with WebSphere Security
- Implementing a Logout Feature
- User Data Constraint
- Configuring a User Data Constraint in RAD
- Summary
- **Lab**
- References

EJB Security

- Overview
- EJB Security
- Setting up EJB Security
- Sample Role Mapping
- Defining Roles
- Setting Method Permission
- Configuring Declarative Security Using RAD
- Defining Roles Using RAD
- Configuring Method Permissions Using RAD
- Disable Security Check
- Disabling Security Check Using RAD
- Excludes List
- Configuring Excludes List Using RAD
- Configuring Unprotected Methods Using WebSphere
- **Lab**
- Programmatic Role-based Security
- Security Role References
- Configuring Security Role Reference Using RAD

Lab

- User Context of a Method Execution
- Accessing User Credentials
- Accessing User Credentials
- Specifying User Context
- Specifying User Context
- Use Caller Identity Scenario
- Run As Scenario
- Configuring Use Caller Identity Using RAD
- Configuring Run As Identity Using RAD
- Mapping Run As Roles to Users Using WebSphere
- WebSphere EJB Delegation Policies
- Configuring Use Identity of Caller Using RAD
- Configuring Use System Identity Using RAD
- Overriding System Identity Using WebSphere
- Configuring Run As Specified Identity Using RAD
- Summary

Lab

- References

SSL Configuration

- Overview
- The Need for Encryption
- Public Key Infrastructure (PKI)
- Certificates
- SSL Basics
- WebSphere and SSL
- WebSphere SSL Configuration
- SSL Configuration Repertoire
- SSL Repertoires
- Creating an SSL Repertoire
- Dummy Certificates
- Key Files
- Trust File
- Default Key Stores
- Obtaining a Certificate
- Key Management Tools
- Using keytool
- Generate a Self-Signed Certificate
- Getting a CA Signed Certificate
- Specify the Key Store
- Different SSL Interactions
- Web Client to Web Server
- Enable SSL For IBM HTTP Server
- Web Server to WebSphere

- Java Client to WebSphere
- Summary
- Review Questions
- Answers
- References

Web Services Security

- Overview
- The Challenges
- Overview of Web Services Security
- WebSphere and Web Services Security
- SOAP Message Security
- Message Integrity
- Message Confidentiality
- Symmetric Encryption Example
- Authentication
- Transport Level Security
- Configuring Security in WebSphere
- Configuring a Server Module
- Configuring a Client Module
- Summary
- Review Questions
- Answers
- References

Security

- Java Security
- Attacks and Dangers
- Overview of JDK Security Features
- Basic Concepts of Computer Security
- Encryption
- Cryptography Algorithm
- Message Digest
- Symmetric Ciphers
- Asymmetric Ciphers
- Digital Signature
- Authentication
- Certificate Manipulation
- Java Cryptography Architecture (JCA)
- Java Cryptography Extension
- Using the MessageDigest Class
- Example of Using the MessageDigest Class
- Using the Signature Class
- Java Security Architecture
- JDK 1.0 Security Model Sandbox
- JDK 1.1 Security Model Trusted Signed Code
- JDK 1.2 Security Model Security Policy
- JDK 1.4 Security Enhancement

- Protection Domains and Security Policies
- ProtectionDomain Class
- Permission Classes
- Using Permission Classes
- Policy Class
- Policy Configuration File
- AccessController Class
- SecurityManager Class
- Using the SecurityManager Class
- Dynamic Class Loader
- Loader Classes
- Example of Security Check in a Class Loader
- Java Security Tools
- Using Java Security Tools Code Signing
- Java Security
- Enabling Java Security
- WebSphere Policy
- WebSphere Policy Files
- Other WebSphere Policy Files
- Application Security
- was.policy
- Using was.policy
- was.policy Example
- Deployment
- Summary
- Review Questions
- Answers
- References

Learning Objectives

- Explain about common SQL Server security threats
- Create and manage SQL logins, Trusted logins, Schemas, Server and Database roles
- Control Execution Context
- Setup Password Policy
- Use the base functionality of SQL Server Policy Based Management
- Audit SQL Server access and configuration changes using SQL Auditing and DDL triggers
- Control and manage processes
- Grant and Revoke all types of SQL Server Permissions
- Manage SQL Server Users and Certificates
- Employ a security architecture set forth for object and data
- Apply CLR level security
- Reduce Surface of Attack
- Create and Control End-points, secure and encrypt end-points
- Setup a Database, Server, and Logon policy for one or more SQL Servers
- Create agent proxies based on a spec.

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for Beginner Administrators who would like to understand baseline Security Standards for Enterprise Database Systems.

Course Description

This hands-on class teaches new SQL Server 2008 System Administrator level students how

to apply security in the different areas of SQL Server 2008. This course teaches how to implement Secure Stored Procedures and View, Secure Tables, Grant and Revoke Database level Permissions, Implement Schema, Create Logins, Roles, and use Windows Groups as the primary logon type, Control Surface of Attack, Create End Points. Students will also learn how to implement basic encryption, and recover databases and objects.

Prerequisites

In order to get the most out of this class, the student should have some experience installing, configuring, and securing Windows Server 2008. Some experience with SQL Server Security and OS Security is helpful, but it's not required.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Explain about common SQL Server security threats
- Create and manage SQL logins, Trusted logins, Schemas, Server and Database roles
- Control Execution Context
- Setup Password Policy
- Use the base functionality of SQL Server Policy Based Management
- Audit SQL Server access and configuration changes using SQL Auditing and DDL triggers
- Control and manage processes
- Grant and Revoke all types of SQL Server Permissions
- Manage SQL Server Users and Certificates
- Employ a security architecture set forth for object and data
- Apply CLR level security
- Reduce Surface of Attack
- Create and Control End-points, secure and encrypt end-points
- Setup a Database, Server, and Logon policy for one or more SQL Servers
- Create agent proxies based on a spec.

TOPICS COVERED IN LECTURE & LAB

Module 1: SQL Server 2008 Security Principles and Concepts

- SQL Server Security Modes
- Creating Trusted Logins
- Creating SQL Logins
- Creating Users based on Domain Groups
- Linking SQL Logins and Database Users
- Understanding SQL Server Level Roles
- Understanding Database Roles
- Setup Password Policy
- Controlling Execution Context

Module 2: SQL Server Schemas as Security Objects

- Overview Schemas
- Creating Schemas
- Adding Users to Schemas

- Granting Rights through Schemas
- Setting a Users Default Schema
- DDL (Data Definition Language) Permissions
- DDL Triggers
- DCL (Data Control Language) Permissions
- Process activation and Kill permissions
- Granting Permissions
- Schemas Best Security Practices

Module 3: Table, View, and Object Security

- DML (Data Manipulation Language) Rights
- Granting and Revoking DML Permissions to tables and views
- Ownership Chains
- Trusted Objects
- Viewing Effective Rights
- Scripting Permissions
- Verifying Permissions
- Securing Surface Area with
- DML Best Security Practices
- Securing Stored Procedures and Functions
- CLR Security

Module 4: Creating and Securing End-Points

- Overview of End-Points
- Defining End-Points
- Securing End-Point ports
- Firewall Settings
- Encrypted End-Points
- Deleting End-Points
- End Point Best Security Practices

Module 5: Disaster Recovery Essentials

- Backup and Restore of System Databases
 - ◆ Master
 - ◆ Model, MDSB, Distribution, Reports, etc.
- Backup and Restore of User Databases
- Offline Backup of SQL Server
- Administrative Account Recovery
- Password Encryption

Module 6: SQL Server Auditing

- Enabling SQL Server Audits
- Reading the Audit Log
- Centralized Auditing
- Auditing Access to Data
- Auditing DDL Changes

- Audit Policy Enforcement
- SQL Audit Best Practices

Module 7: Policy Based Management

- Controlling Security with Policy Bases Management
- Setup a Database Policy
- Setup a Server Policy
- Setup a Storage Policy
- Setup a SQL Compatibility Level Policy
- Enforce Compliancy
- Recommended Database and Server Settings

Learning Objectives

2010 SECURITY SERIES

- Apply general SQL Server Hardening and Best security practices
- Identify the vulnerabilities and defenses for common SQL Server Security attacks like SQL injection
- Define a security Architecture for enterprise wide SQL Server environments
- Lock down and minimize the surface of attack
- Establish a security baseline and apply that set configurations to 1 or more servers using various policies levels of SQL Server Policy Based Management
- Establish a monitored compliancy system for the enterprise security across all new and deployed SQL Server instances
- Determine the best authentication to be used for various scenarios
- Establish Password and Logon policies
- Enforce configuration standards
- Locate and remediate servers that fall outside of the security standards set forth
- Apply SQL Server best security practices
- Employ various levels of encryption for objects and data
- Create a secure by default installation package, and secure in deployment configuration set
- Apply Policy Based Management for all security requirements
- Backup, Restore and patch instances and databases
- Control Execution Context and use Agent proxies to reduce the privileges available to the SQL Agent account
- Audit all facets of SQL Server Configuration and operation

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to

change without notice.

Course Audience

This course is appropriate for Intermediate to Advanced Database Administrators who would like to implement Security Standards and Best Practices for Enterprise Database Systems.

Course Description

This hands-on class teaches DBA level students how to harden and secure SQL Server 2008 new deployments, and existing installations in an enterprise environment. This course teaches how to implement Policy Based Management to create conditions, facets, policies, and how to use policies to ensure compliance, setup transparent encryption, and run security baseline checks against one or more SQL Server 2008 instances. This class is primarily focused on the securing the SQL Server 2008 Database Engine, Agent Services, End Points, SQL objects, SQL data, and SQL metadata. Each Module has its own lab so that the student gets to apply what they have learned in each module.

Prerequisites

In order to get the most out of this class, the student should have some experience installing, configuring, and securing SQL Server 2008. Some experience with Active Directory is helpful since Group Policy Objects are used to secure most SQL Server enterprise environments, but it's not required. Students should also have a solid understanding of how the SQL Server Agent and SQL Agent Jobs operate.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Apply general SQL Server Hardening and Best security practices
- Identify the vulnerabilities and defenses for common SQL Server Security attacks like SQL injection
- Define a security Architecture for enterprise wide SQL Server environments
- Lock down and minimize the surface of attack
- Establish a security baseline and apply that set configurations to 1 or more servers using various policies levels of SQL Server Policy Based Management
- Establish a monitored compliancy system for the enterprise security across all new and deployed SQL Server instances

- Determine the best authentication to be used for various scenarios
- Establish Password and Logon policies
- Enforce configuration standards
- Locate and remediate servers that fall outside of the security standards set forth
- Apply SQL Server best security practices
- Employ various levels of encryption for objects and data
- Create a secure by default installation package, and secure in deployment configuration set
- Apply Policy Based Management for all security requirements
- Backup, Restore and patch instances and databases
- Control Execution Context and use Agent proxies to reduce the privileges available to the SQL Agent account
- Audit all facets of SQL Server Configuration and operation

TOPICS COVERED IN LECTURE & LAB

Module 1: Overview of SQL Server Security Best Practices (Most topics are covered in detail with later modules)

- SQL Server Hardening
 - ◆ Physical Security
 - ◆ Network Access
 - ◆ Vulnerabilities
 - SQL Injection Defense
 - Remote Execution Defense
 - Privilege Elevation Defense
 - ◆ Overview of Auditing
 - ◆ Surface Area Reduction
 - ◆ Service Account Selection and Management
 - ◆ Authentication Modes
 - ◆ Network Connectivity
- Database Engine Security Facets
 - ◆ Lockdown of System Stored Procedures
 - ◆ Password Policy
 - ◆ Administrator Privileges
 - ◆ Database Ownership and Trust
 - ◆ Schemas
 - ◆ Authorization
 - ◆ Catalog Security
 - ◆ Remote Data Source Execution
 - ◆ Execution Context
- Encryption of Data and Objects (Stored Procedures, Triggers, Views, etc)

- Microsoft TWC (Trustworthy Computing objectives)
- Microsoft Baseline Security Analyzer
- SQL Server Best Practices Analyzer
- Patching SQL Server 2008

Module 2: Security Features and Encryption in SQL Server 2008

- Encrypting Sensitive Data
 - ◆ Built-in Cryptography Hierarchy
 - ◆ Using Database Encryption Key's (DEK)
 - ◆ Transparent Data Encryption
 - ◆ Extensible Key Management
- Hardware Security Modules
- High-performance, granular auditing
- Configure "Secure by Default" and "Secure in Deployment" Configurations

Module 3: SQL Server Policy Based Management

- Overview Policy Based Management
- Create Policies, Conditions, and Rules
- Creating and Assigning Multiple Policies
- Managing Security for Multiple Servers with Policy Based Management
- Enforcing SQL Server Policy Based Management
- Configure the surface area with automated Policy-Based Management
- Ensuring Enterprise Compliance with Configuration Policies for:
 - ◆ Servers
 - ◆ Databases
 - ◆ Database Objects
- Reducing Exposure to Security Threats:
 - ◆ New Surface Area facet
 - ◆ Controlling Services and Features

Module 4: Increasing Control over the SQL Server Agent Services and Jobs

- MDSB Fixed Database Roles
- Increasing Control Over Agent Services
- SQL Server Agent Principles and Proxies
- Using Multiple Proxy Accounts
- Securely Executing SSIS Packages
- Securely Executing Jobs on other Servers
- Controlling and Applying Execution Context
- Preventing SQL Injection Attacks

Module 5: Enforcing Password and Login Policies

- Review SQL Server Authentication Types, Logins, Users, Roles
- Automatically Apply Password Policies for SQL logins
 - ◆ Periodic password change
 - ◆ Min length
 - ◆ Max length
 - ◆ Complexity/Character combinations
- Inheriting Active Directory Password and Account Policies
- Overriding Active Directory Password and Account Policies
- Scripting Secure SQL Logins

Module 6: SQL Server 2008 Security Compliance

- Methods for Checking and Enforcing Compliance
- Using Windows Update to Apply SQL Server Security Patches
- Applying Software Updates
- Ensuring Security Baseline Compliance and Logging Results
- Provide security enhanced metadata access
- Enhance security features with execution context
- Signing Code Modules (Digital Signatures)

Module 7: SQL Server 2008 Auditing

- Auditing Database Activity
- Enhanced auditing with the SQL Server Audit New
- Define audits to automatically record activity
 - ◆ Text Log Files
 - ◆ Windows Application log
 - ◆ Windows Security log

Module 8: Review Best Practices, Conclusions, Q&A (via WebEx for Online Deliveries)

- SQL Server Hardening
 - ◆ Physical Security
 - ◆ Network Access
 - ◆ Vulnerabilities
 - ◆ Surface Area Reduction
 - ◆ Service Account Selection and Management

- ◆ Authentication Modes
- ◆ Network Connectivity
- ◆ Policy Based Management
- ◆ Encryption
- ◆ Password Policy
- ◆ Ownership and Trust
- ◆ Remote Data Source Execution
- ◆ Execution Context

Learning Objectives

- Identify security requirements for different types of information
- Perform a security risk assessment of a SQL Server application
- Understand 10 best practices for secure SQL Server development. Some of the best practices are implemented on the SQL side, and some on the client side.
- Fix Security holes
- Follow best practices for future development projects
- Identify exceptions to the security plan
- Determine what information needs to be encrypted

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for Intermediate SQL Developers who would like to understand baseline Security Standards for Enterprise Database Systems.

Course Description

This 24 hour hands-on class teaches SQL Server developers and application developers how to determine if their application meets security requirements and follow best practices for building and updating secure SQL Server applications. The Student will start off with a non secure ASP.net / SQL Server Application. After identifying the client side, network, and SQL Server back end vulnerabilities, the developers will get to work security all aspects of the application. The students will then

determine what special handling requirements (IE encryption) need to be enforced for the data. Once the application is secured using best practices, the developer will implement a security governance plan with programs, policies, and auditing that ensures the applications will stay in compliance. The goal is to secure SQL Server applications, with minimal impact on the UI elements of the application. Students will be able to determine those requirement from application to application and implement the changes needed on a per application basis.

Prerequisites

In order to get the most out of this class, the student should have some experience with ADO.NET, ASP.NET, Visual Studio, .NET development, IIS, SQL Server programming, application development and testing, Windows Authentication, SQL Server security (logins and users), and TSQL. Some experience creating and managing databases with SQL Server Management Studio is recommended but not required.

Course Objectives

At course completion the student will be able to perform the following tasks:

ke all types of SQL Server Permissions

- Identify security requirements for different types of information
- Perform a security risk assessment of a SQL Server application
- Understand 10 best practices for secure SQL Server development. Some of the best practices are implemented on the SQL side, and some on the client side.
- Fix Security holes
- Follow best practices for future development projects
- Identify exceptions to the security plan
- Determine what information needs to be encrypted

- ◆ Performing a manual SQL Injection attack
- ◆ Running automated scans/attacks
 - SQLMap
 - Nikto
- Analyzing vulnerable areas of the web application
- Create a plan to eliminate or mitigate vulnerabilities
 - ◆ Plan for securing the client connections
 - ◆ Plan for securing the SQL Server database
 - ◆ Plan for securing the ISS connections
 - Client to IIS
 - IIS to SQL Server
 - ◆ Plan to identify the information which requires access auditing
 - ◆ Plan to identify the information which requires encryption
- Understand SQL Server Authentication Types
 - ◆ SQL Server Authentication

TOPICS COVERED IN LECTURE & LAB

Module 1:

Overview of SQL Server Security Facets

- SQL Server Hardening
 - ◆ Physical Security
 - ◆ Network Access
 - ◆ Introduction to common Vulnerabilities
 - SQL Injection Defense
 - Remote Execution Defense
 - Privilege Elevation Defense
 - ◆ Overview of Auditing
 - ◆ Understanding Surface Area Reduction
 - ◆ Service Account Selection and Management
 - ◆ Authentication Modes
 - ◆ Network Connectivity
- Overview Application framework and code hardening
 - ◆ ADO.NET commands and Parameters
 - ◆ Data Transmission security
 - ◆ Session Security
 - ◆ Avoiding the use of Dynamic Queries and Strings
 - ◆ Implementing Stored Procedures for ADO.NET client calls
 - ◆ Understanding Kerberos
 - ◆ Understanding Single Sign on

Module 2:

Testing Applications for SQL Injection Vulnerabilities and Planning Remediation

- Performing a SQL injection attack on an insecure web application

Module 3:

Creating a SQL Injection Remediation Plan

- Reviewing code for
 - ◆ Dynamic SQL
 - ◆ Stored credentials
 - ◆ Connection vulnerabilities
 - ◆ Transmission of user credentials in clear text
 - ◆ Transmission of confidential data in clear text
 - ◆ Non validated fields in the client application
- SQL Injection Remediation
 - ◆ Identify Stored Procedures which need to be created
 - ◆ Identify Views which need to be created
 - ◆ Identifying ADO.NET parameters and store procedure calls
 - ◆ Identify affected areas of the application and UI elements
- Understanding secure connections
 - ◆ Kerberos
 - ◆ Single Sign on
- Authentication layers
 - ◆ Understanding the authentication layer for client to IIS
 - ◆ Understanding the authentication layer for IIS to SQL Server
 - ◆ Understanding Kerberos delegation
- Preparing a plan of action based on discovered vulnerabilities

Module 4: Implementing a SQL Injection Remediation Plan

- Creating and testing the required stored procedures on backend
- Implementing check constraints on the database
- Replacing the dynamic SQL code with calls to stored procedures
- Testing the client the application for functionality
- Re-testing to verify SQL injection vulnerability has be eliminated
- Implement SQL Server validation
- Implement client side validation
- Come up with a testing plan for the ASP.NET application

Module 5: Securing the SQL Server Data

- Methods for Checking and Enforcing Compliance
- Using Widows Update to Apply SQL Server Security Patches
- Applying Software Updates
- Ensuring Security Baseline Compliance and Logging Results
- Provide security enhanced metadata access
- Enhance security features with execution context
- Signing Code Modules (Digital Signatures)
- Understanding security compliance
 - ◆ Sarbanes Oxley (SOX)
 - ◆ Health Insurance Portability and Accountability Act (HIPAA)
 - ◆ C2 audit specification
 - ◆ Programs
 - ◆ Policies
- Determining which parts of the database should be encrypted
- Understanding the principal of least user privilege (LUP)
 - ◆ Implementing SQL Server Logins
 - ◆ Implementing SQL Server Users
 - ◆ Planning for Security Roles
 - Implementing SQL Server roles
 - ◆ Forcing Windows Authentication
 - ◆ Nesting users and roles
- Implementing Least User Privilege

- ◆ Granting permissions to execute stored procedures
- ◆ Granting permissions to views
- ◆ Granting Permissions to Tables
- Testing for excessive permissions
- Setting up the backend database to meet security needs and follow best practices

Module 6: Enforcing Password and Login Policies

- Review SQL Server Authentication Types, Logins, Users, Roles
- Automatically Apply Password Policies for SQL logins
 - ◆ Periodic password change
 - ◆ Min length
 - ◆ Max length
 - ◆ Complexity/Character combinations
- Inheriting Active Directory Password and Account Policies
- Overriding Active Directory Password and Account Policies
- Scripting Secure SQL Logins
- Changing Execution Context
- Setting the ASP.NET application for secure authentication

Module 7: Securing the Network communications

- Understanding SQL Connection types
 - ◆ .NET frameworks connection account
 - ◆ OLEDB
 - ◆ ODBC
 - ◆ SQL Server Ports
 - ◆ SQL Server End Points
 - ◆ Encrypting Connections
 - ◆ Performing Network Captures of SQL Server Traffic
- Evaluation of the authentication process
 - ◆ Testing for encrypted passwords and credentials
 - ◆ Testing for clear text confidential data
- Introduction to Public Key Infrastructure
 - ◆ Symmetric Keys
 - ◆ Asymmetric Keys
- Deploying an SSL certificates to web server pages
 - ◆ Set secure pages for secure socket layer (https)
- Understanding Internet Protocol Security (IPSEC)
 - ◆ Implement IPSEC connections

- Understanding SQL Server end points
 - ◆ Implementing an encrypted Endpoint for communications
- Implementing a network security plan for the application

Module 8: SQL Server Encryption

- Encrypting Sensitive Data
 - ◆ Built-in Cryptography Hierarchy
 - ◆ Using Database Encryption Key's (DEK)
 - ◆ Transparent Data Encryption
 - ◆ Extensible Key Management
- Hardware Security Modules
- Encrypting specific columns and rows of data
- Encrypting entire tables of data
- Encryption of Data and Objects (Stored Procedures, Triggers, Views, etc)
- High-performance, granular auditing
- Configure "Secure by Default" and "Secure in Deployment" Configurations
- Implementing SQL encryption for the storage of confidential data

Module 9: Handling security requirements with SQL Server Policies

- Overview Policy Based Management
- Create Policies, Conditions, and Rules
- Creating and Assigning Multiple Policies
- Managing Security for Multiple Servers with Policy Based Management
- Enforcing SQL Server Policy Based Management
- Configure the surface area with automated Policy-Based Management
- Ensuring Enterprise Compliance with Configuration Policies for:
 - ◆ Servers
 - ◆ Databases
 - ◆ Database Objects
- Reducing Exposure to Security Threats:
 - ◆ New Surface Area facet
 - ◆ Controlling Services and Features
- Implementing best practices with SQL Server Policy based management

Module 10: Increasing Control over the SQL Server Agent Services and Jobs

- MDSB Fixed Database Roles
- Increasing Control Over Agent Services
- SQL Server Agent Principles and Proxies
- Using Multiple Proxy Accounts
- Securely Executing SSIS Packages
- Securely Executing Jobs on other Servers
- Controlling and Applying Execution Context
- Preventing SQL Injection Attacks during ETL

Module 11: SQL Server 2008 Auditing

- Auditing Database Activity
- Enhanced auditing with the SQL Server Audit New
- Define audits to automatically record activity
 - ◆ Text Log Files
 - ◆ Windows Application log
 - ◆ Windows Security log

Module 12: Review 10 Security Best Practices for SQL Server Development

- Avoid using canned SQL server roles (Db_datareader, Db_datawriter) when the login does not need read or write access to all data
- Eliminate AD_HOC (dynamic) SQL queries from the applications
- Eliminate unauthorized disclosure of sensitive data by using view and stored procedures
- Do not use guest or public roles for permissions
- Use parameterized queries and stored procedures for all SQL calls and validate the data on the client form and server database.
- Use Window Authentication whenever possible.
- Encrypt sensitive data in transit (IPSEC, SSL, Encrypted End Points)
- Encrypt sensitive data at rest with Asymmetric keys with key lengths great than 1024.

- Do use XP_CMD shell stored procedures
- Implement SQL Server Agent Proxies:
 - ◆ For external calls
 - ◆ SSIS packages
 - ◆ Agent Jobs

Learning Objectives

2010 SECURITY SERIES

- Create and Manage all type of users, local groups, and domain groups
- Apply ACL's to objects and resources
- Configure and Troubleshoot User Account Control (UAC)
- Understand how Kerberos tokens are elevated
- Understand NTLM versus Kerberos/LDAP authentication works
- Apply and enforce a local security policy
- Force a domain wide policy down to servers from active directory
- Monitor and evaluate security policy compliancy across a group of servers
- Implement a secure file and share infrastructure using recommended best practices
- Secure IIS objects and content
- Enable End-point and server based protection against malware, viruses, and other threats
- Apply security best practices for auditing, sharing, printing, patch management
- Employ baseline configuration management

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for Beginner Administrators who would like to understand baseline Security Standards for Enterprise Server 2008 Systems.

Course Description

This hands-on class teaches new Windows Server 2008 System Administrator level stu-

dents how to apply security in the different areas of Windows Server 2008. This course teaches how to implement Group Policies, Services, Roles, Features, using Server Manger, secure file system objects and shares, understand share level security, protect the registry, configure and troubleshoot group policy, secure IIS resources, secure DNS, encrypted data and network traffic, backup /restore system configurations.

Prerequisites

In order to get the most out of this class, the student should have some experience installing, configuring, and securing Windows Server 2008. Some experience with Active Directory is helpful since Group Policy Objects are used to secure most Windows Server enterprise environments, but it's not required.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Create and Manage all type of users, local groups, and domain groups
- Apply ACL's to objects and resources
- Configure and Troubleshoot User Account Control (UAC)
- Understand how Kerberos tokens are elevated
- Understand NTLM versus Kerberos/LDAP authentication works
- Apply and enforce a local security policy
- Force a domain wide policy down to servers from active directory
- Monitor and evaluate security policy compliancy across a group of servers
- Implement a secure file and share infrastructure using recommended best practices
- Secure IIS objects and content
- Enable End-point and server based protection against malware, viruses, and other threats
- Apply security best practices for auditing, sharing, printing, patch management
- Employ baseline configuration management

TOPICS COVERED IN LECTURE & LAB

Module 1: Windows Server 2008 Security Principles and Concepts

- Creating Local Users
- Creating Domain Users
- Creating and Using Local Groups to apply security
- Creating and Using Domain Groups to apply security
- User Account Control
- Understanding Token Elevation
- Understanding NTLM Authentication
- Understanding Kerberos/LDAP Authentication
- Services Account Management and Managed Service Accounts (MSO's)

Module 2: Security Policy

- Local Security Policy
- Domain Security Policy

- Modifying Local Security Policy
- Setting Domain Security Policy with Group Policy Objects
- Active Directory Users and Computers
- Deploying Group Policy Objects
- Troubleshooting Group Policy
- Implementing Administrative Templates
- Implementing Security Templates
- Managing Account Policy with GPO's
- Understanding Group Policy Inheritance and override

Module 3: File System and Share Security

- File System Security Rights
- Folder versus File rights
- Local versus Share Permission
- Granting, Revoking, Denying rights
- Denied Access
- Rights versus Privileges
- Understanding UAC (User Account Control)
- Troubleshooting UAC
- Administrative Shares

Module 4: Securing Objects

- Active Directory Object Security
- Registry Security
- IIS Content Security
- Controlling Desktop with Group Policy
- Securing Printers
- Securing OS configuration

Module 5: Windows Updates

- Controlling Updates
- Configuring Updates
- Rolling back a configuration
- Malware Protect
- Endpoint Protection
- Antivirus Protection

Module 6: Review / Q&A

- Windows Server 2008 Security Principles and Concepts
- Security Policy
- File System and Share Security
- Windows Updates
- Securing Objects

Learning Objectives

2010 SECURITY SERIES

- Apply general Server hardening and security best practices for Windows Server 2008
- Perform a full assessment of security risks and vulnerabilities for a Windows Server 2008 Server infrastructure
- Setup a security Deployment Lifecycle (SDL) for the server or servers
- Create and Deploy security templates to servers via group policy
- Perform threat modeling to establish vulnerabilities in a given scenario
- Use MBSA throughout the enterprise to report possible configuration settings that jeopardize security
- Establish an update and patch management system
- Apply NAP, and advanced firewall settings
- Implement IPSEC and monitor traffic
- Implement Server and AD auditing
- Create and use Read-Only domain controllers
- Monitor group policy application
- Setup an Enterprise Security Architecture (ESA)
- Create and Monitor Security Baselines
- Setup PKI and deploy certificates for various applications
- Identify and mitigate various application security vulnerabilities
- Control Authentication types
- Lock down IIS
- Implement Disaster Recovery

This course description should be used to determine whether the course is appropriate for you based on your current skill and technical training needs. Technical information is provided on the intended audience, course prerequisites, and covered topics. Course content, prices, and availability are subject to change without notice.

Course Audience

This course is appropriate for students whom have Intermediate to Advanced experience with Windows Server 2008.

Course Description

This hands-on class teaches Windows Server 2008 System Administrator level students how to harden and secure Windows Server 2008 new deployments, and existing installations in an enterprise environment.

This course teaches how to implement Policies, Services, and monitoring/remediation methods to confidently secure Windows Server 2008 in the enterprise. This class is primarily focused on the securing the Windows Server 2008 core technologies like TCP/IP ports, services, patching, networking, certificate based encryption and signatures. Students will also learn the best practices for maintaining a secure and recoverable Windows Server 2008 environment.

Prerequisites

In order to get the most out of this class, the student should have some experience installing, configuring, and securing Windows Server 2008. Some experience with Active Directory is very helpful since Group Policy Objects are used to secure most Windows Server enterprise environments, but it's not required.

Course Objectives

At course completion the student will be able to perform the following tasks:

- Apply general Server hardening and security best practices for Windows Server 2008
- Perform a full assessment of security risks and vulnerabilities for a Windows Server 2008 Server infrastructure
- Setup a security Deployment Lifecycle (SDL) for the server or servers
- Create and Deploy security templates to servers via group policy
- Perform threat modeling to establish vulnerabilities in a given scenario
- Use MBSA throughout the enterprise to report possible configuration settings that jeopardize security
- Establish an update and patch management system
- Apply NAP, and advanced firewall settings
- Implement IPSEC and monitor traffic
- Implement Server and AD auditing
- Create and use Read-Only domain controllers
- Monitor group policy application

- Setup an Enterprise Security Architecture (ESA)
- Create and Monitor Security Baselines
- Setup PKI and deploy certificates for various applications
- Identify and mitigate various application security vulnerabilities
- Control Authentication types
- Lock down IIS
- Implement Disaster Recovery

TOPICS COVERED IN LECTURE & LAB

Module 1: Overview Windows Server 2008 Security Best Practices

- Assessment of Risks
- Setting Lockout to prevent Password Attacks
- Specialized Security - Limited Functionality (SSLF) environment
- Preventing Access to Registry Editing Tools
- Security Development Lifecycle (SDL)
- Threat modeling
- Deploying Security Templates via Group Policy
- Run MBSA against Servers and Clients
- Security Updates, Patches, Service Pack deployments

Module 2: Security features in Windows Server 2008

- Security policies enforcement in Windows Server 2008?
- New Windows Firewall Features
- IPSEC
- Network Location Awareness (NLA)
- Network Access Protection (NAP)
- Health Policy Validation
- Health Policy Compliance
- Limited Access Security
- Windows Firewall Advanced Security (WFAS)
- Windows Server Hardening
- Server and Domain Isolation
- Active Directory Auditing
- Read Only Domain Controllers
- Bit Locker Drive Encryption
- Removable Device Installation Control

- Administrator Role Separation
- Server Core Installation

Module 3: Creating a Security Policy for Windows Server 2008 Servers

- Creating and Implementing Security Templates
- Deploying Security Templates with Group Policy
- Monitor Group Policy Execution
- MBSA – Microsoft Baseline Security Analyzer
- System Center Configuration Manager
 - ◆ Software Update Point and Software Updates Deployments
 - ◆ Desired Configuration Management (DCM)
 - ◆ Compliance and Remediation
- Windows Server 2008 Security Compliance Manager
- Implementing a Patch Management and Software Update Policy

Module 4: Monitoring Security Policy Baselines for Windows Servers and Clients

- Understanding Threats and Counter Measures
- Installing and Configuring Security Baseline Compliance Manager (Formerly Security Baseline Tool).
- Creating and Importing Baselines for Windows Server 2008, Windows Server 2003, Windows XP, Windows Vista, Windows 7
- Configuring Baselines for Desktop, Computer, IE 7 and IE 8, Domain, Domain Controller, Member Server, Client.
- Third Party Baselines

Module 5: Enterprise PKI (Public Key Infrastructure)

- Setup and Configure Root CA
- Working with Certificates
- Issue Web Server Certificates
- Create and Issue Web Server Certificates
- Create and Issue EFS Certificates
- Certificate Revocation
- Easier Management through PKI-View
- Certificate Web Enrollment
- Certificate Enrollment through Group Policy

- Network Device Enrollment Service.
- Certificate Policy Settings.
- Certificate Deployment changes.
- Online Certificate Status Protocol (OCSP) support.
- Managing Certificates with Group Policy.
- Cryptographic Next Generation

Module 6: Application Vulnerabilities and Remediation

- Microsoft Security Bulletins
- Microsoft Windows Server Service Could Allow Remote Code Execution
- Microsoft SMB Remote Code Execution Vulnerability
- Microsoft MSDTC and COM+ Remote Code Execution Vulnerability
- Microsoft Windows DCO0M RPCSS Service Vulnerabilities
- Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability
- Microsoft IIS FTP Server Remote Stack Based Overflow
- Writeable SNMP Information
- Adobe Flash Player Vulnerabilities
- Adobe Flash Player Update to Address Security Vulnerabilities 116244
- Adobe Acrobat and Adobe Reader Vulnerabilities
- Adobe Reader JavaScript Methods Memory Corruption Vulnerability
- Sun Java Vulnerabilities
- Microsoft Office PowerPoint Could Remote Code Execution Vulnerability
- Microsoft Excel Remote Code Execution Vulnerability
 - ◆ Microsoft Word Multiple Remote Code Execution Vulnerabilities
 - ◆ WordPad and Office Text Converters Remote Code Execution Vulnerability
 - ◆ Vulnerabilities in Microsoft DirectShow
- General MS Office 2007 and 2010 Security Guidelines

Module 7: WSUS (Windows Server Update Services)

- Updates for Windows, Office, Exchange Server, and SQL Server, with additional product support over time

- Specific updates can be set to download automatically
- Automated actions for updates determined by administrator approval
- Ability to determine the applicability of updates before installing them
- Targeting
- Replica synchronization
- Reporting
- Extensibility

Module 8: Hardening IIS

- Best Practices for IUSR & IWAM accounts
- General Assumptions
- Security Design Guidelines
- Authentication Types
- SSL
- IIS Lockdown
- URLScan
- Logging
- Recovery
- Installation Configuration
- Deleting Default Content
- Backup and Restore IIS meta-base
- IPSEC filters

Module 9: Server Disaster Recovery

- File System Backup and Recovery
- Active Directory Backup and Recovery
- System State Backup and Recovery
- Application Server Backup and Recovery

Module 10: Security Best Practices Review

- Minimizing Surface of Attack
- Auditing
- Baseline Configuration and Enforcement
- Monitoring for Compliance
- Remediation
- Documentation
- Patch Management
- Disaster Recovery Planning