DEVELOPER 544

Hands On | Four Days | Laptop Required | 24 CPEs

DEV544: Secure Coding in .NET: Developing Defensible Applications

ASP.NET and the .NET framework have provided web developers with tools that allow them an unprecedented degree of flexibility and productivity. On the other hand, these sophisticated tools make it easier than ever to miss the little details that allow security vulnerabilities to creep into an application. Since ASP.NET, 2.0 Microsoft has done a fantastic job of integrating security into the ASP.NET framework, but the responsibility is still on application developers to understand the limitations of the framework and ensure that their own code is secure.

Have you ever wondered if the built-in ASP.NET validation is effective? Have you been concerned that WCF web services might be introducing unexamined security issues into your application? Should you feel uneasy relying solely only on the security controls built into the ASP.NET framework? Secure Coding in .NET will answer these questions and far more.

What Does the Course Cover?

This is a comprehensive course covering a huge set of skills and knowledge. It's not a high-level theory course. It's about real programming. In this course you will examine actual code, work with real tools, build applications, and gain confidence in the resources you need for the journey to improving the security of .NET applications.

Rather than teaching students to use a set of tools, we're teaching students concepts of secure programming. This involves looking at a specific piece of code, identifying a security flaw, and implementing a fix for flaws found on the OWASP Top 10 and CWE/SANS Top 25 Most Dangerous Programming Errors.

The class culminates in a Secure Development Challenge where you perform a security review of a real-world open-source application. You will conduct a code review, perform security testing to actually exploit real vulnerabilities, and finally, using the secure coding techniques that you have learned in class, implement fixes for these issues.

PCI Compliance

Section 6.5 of the Payment Card Industry (PCI) Data Security Standard (DSS) instructs auditors to verify that processes exist that require training in secure coding techniques for developers. If your application processes cardholder data and you are required to meet PCI compliance then this course is for you.

BE SECURE
BEFORE
YOU'RE NEXT

Who Should Attend

This course is intended for:

- ASP.NET developers who want to build more secure web applications
- .NET framework developers
- Software engineers
- Software architects
- Developers who need to be trained in secure coding techniques to meet PCI compliance

This class is focused specifically on software development, but it is accessible enough for anyone who's comfortable working with code and has an interest in understanding the developer's perspective. This could include:

- Application security auditors
- Technical project managers
- Senior software QA specialists
- Penetration testers who want a deeper understanding of how to target ASP.NET web applications or who want to provide more details vulnerability remediation options

You Will Be Able To

- Use a web application proxy to view and manipulate HTTP requests and responses.
- Review and perform basic exploits of common .NET web application vulnerabilities, such as those found in the SANS/CWE Top 25 and the OWASP Top 10:
  - Cross-Site Scripting
  - Parameter Manipulation
  - Open Redirect
  - SQL Injection
  - Session Hijacking
  - Clickjacking
  - Cross-Site Request Forgery
  - Man-in-the-middle (MITM)
- Mitigate common web application vulnerabilities using industry best practices in the .NET framework, including the following:
  - Input Validation
  - Blacklist & Whitelist Validation
  - Regular Expressions
  - Command Encoding
  - Output Encoding
  - Content Security Policy
  - Client-side Security Headers
- Understand built-in ASP .NET security mechanisms, including the following:
  - Event Validation
  - Request Validation
  - View State
  - Entity Framework
  - Forms Authentication
  - Membership Provider
- WCF
- Apply industry best practices (NIST, PCI) for cryptography and hashing in the .NET framework.
- Implementing a secure software development lifecycle (SDLC) to include threat modeling, static analysis, and dynamic analysis

544.1 Data Validation

Improper data validation is the root cause of the most prevalent web application vulnerabilities today. Beginning on the first day, you will learn about some of the most prevalent web applications vulnerabilities such as XSS, SQL Injection, Open Redirects, and Parameter Manipulation. You will see how to find these issues and how to recreate them in a running application. Then you will use a variety of methods to actually fix these vulnerabilities in your C# code. The course is full of hands on exercises where you can apply practical data validation techniques that you can use to prevent common attacks with defense ranging from input validation, output encoding, and use of new techniques like Content Security Policy.

Topics: Web Application Attacks; Web Application Proxies; Parameter Manipulation; Cross-Site Scripting (XSS); Open Redirect; SQL Injection; HTTP Response Splitting; Input Validation; Indirect Selection; Blacklists; Whitelists; Regular Expressions; Event Validation; Character Encoding; Command Encoding; Content Security Policy; LINQ & Entity Framework

544.2 HANDS ON: Authentication and Session Management

A secure architecture is critical for mission critical .NET applications. You will learn about various built-in .NET security features such as cryptography, password storage, web service security and many other .NET features you should consider while writing secure code. A number of hand-on exercises will guide you through writing a cryptography utility for storing sensitive data and user passwords, protecting data in memory, exploiting a running application using DLL Injection, and much more.

Topics: Authentication Factors; Authentication Attacks; Authorization Attacks; Password Management; Basic, Digest, & Windows Authentication; Forms Authentication & Membership Provider; Race Conditions; Session Identifiers; Man-in-the-middle (MITM) Attacks; Cross-Site Request Forgery (CSRF); Clickjacking; Session Hijacking; Session Fixation; Session Management; Cookie Security; ViewState

544.3 HANDS ON: Secure .NET Architecture

Understanding how to leverage .NET to design a secure architecture with solid secure coding principals is critical to application security. This course combines tried and tested information security principals with secure coding principals to help you build rock solid applications.

Topics: Cryptography; Password Storage; Threading; String Immutability; Numeric Overflow; Risks of Malicious Code; Exception Handling; Auditing and Logging; Web Service Security

544.4 HANDS ON: Secure Software Development Lifecycle

We will take a look at each phase of the SDLC and discuss how security fits into the process. Using what you have learned about Web application vulnerabilities, you will get the opportunity to review code from an open source application to identify various vulnerabilities. Then, you will then perform security testing and actually exploit these weaknesses. Once they have been exploited, you will then fix them using the security coding techniques you have learned in class.

Topics: Security Training; Security Requirements; Secure Design; Threat Modeling; Implementation; Static Analysis; Peer Reviews; Secure Code Review; Verification; Dynamic Analysis; Penetration Test Reports; Release; Response

"DEV544 does a terrific job at discussing security in .net, a fairly elusive part of .net programming."

-Craig Allyn Moore, Oncology Nuring Society

DEV544 Training Formats

(subject to change)

Live Training
sans.org/security-training/by-location/all

Community SANS
sans.org/community

OnSite
sans.org/onsite

OnDemand
sans.org/ondemand

SelStudy
sans.org/selfstudy

"I'm a repeat student at SANS conference. Always great courses and knowledgeable instructors"

-Linh Sithihao, USTW