Ingraining security into the mind of every developer.

Learn how to avoid security issues up front instead of fixing problems following design, development, or deployment. Train and certify your teams and contractors in the core competencies of secure application development.

software-security.sans.org
Dear Colleague,

News of data breaches, corporate hacks, and cyber crime top the headlines. It is widely accepted that many of these threats are the result of insecure software. Security should not be an afterthought for software that we rely on every day to power our websites, businesses, and other critical infrastructure. It is our trust in these applications that enables organizations and customers to operate and interact.

Software development is an involved process, with many stakeholders throughout the software development life cycle (SDLC). Everyone involved in creating software—from developers, architects, testers, and managers to security professionals—should have some knowledge about current security threats and the need for software assurance and security. Application security best practices are now the minimum standard of due care for the delivery of critical applications that are essential to business, personal data, and our national security.

To help you build secure software and applications that are resistant to attacks, SANS has created an Application Security Curriculum with classes in secure coding, web application security, pen testing and ethical hacking, and software security awareness. These courses are written and taught by world-class instructors who are also everyday practitioners working on building secure applications and defending them against the same threats you face!

SANS also has a number of free resources that cover these very threats. There are many great ways to stay informed and get involved. So take advantage of SANS’ Application Security training and resources. I hope to see you online or at an upcoming SANS event!

Sincerely,

Frank Kim,
Software Security Curriculum Lead
@sansappsec

Software Security Curriculum

The SANS AppSec curriculum features courses for developers as well as security professionals who want to master the practical steps necessary to defend applications, systems, and networks against the most dangerous threats. The courses are intensive, immersion training full of immediately useful techniques. They were developed through a consensus process involving hundreds of developers, architects, administrators, security managers, and information security professionals. They address secure coding principles, security fundamentals and awareness, and the in-depth technical aspects of the most crucial areas of application security, secure coding, and IT security.

NOTE:
These courses cover the CWE/SANS Top 25 as well as the OWASP Top 10.
There’s always a seat open with SANS

ONLINE TRAINING for Developers

Whether you want to learn via:

- Online Classrooms → vLive
- E-Learning Tool → OnDemand
- Live Event Interaction → Simulcast
- or On Your Own → SelfStudy

An online training option is ready for you.

*Online computer-based training allows you to:*

- Gain security knowledge when it is most convenient for you
- Learn at your own pace and schedule
- Get timely answers to your most pressing security questions
- Complete training in short sessions to fit your work schedule
- Reduce travel costs and travel time

*Our online courses make it easy for you to:*

- Check your knowledge with quiz questions
- Track progress using our reporting dashboard
- Review the content whenever you want

All of our online courses are updated annually to cover the latest threats, tools, tips, and tricks!
Critical software systems designed to enable business functions are often at the root of many headlines about data breaches and corporate hacks. The most common attacks are often caused by simple mistakes that occur while software is being developed and deployed.

By educating everyone involved in the software development process, including developers, architects, managers, testers, business owners and partners, ensure that your team can properly build defensible applications from the start and reduce the chances that your organization will fall victim to one of today’s data security threats. STH.Developer provides the pinpoint software security awareness training that your team needs when they need it most. And they can learn these skills working from their own desks!

The first set of modules utilizes the Open Web Application Security Project’s (OWASP) Top 10 web vulnerabilities. This framework, a proven and validated industry standard for secure web development, consists of an in-depth series of modules that provide the developer with clear technical information about these security vulnerabilities and how to prevent them within their own code.

The following are the modules available for the STH.Developer computer-based training program.

- **MODULE 00**: Introduction
- **MODULE 01**: Injection Flaws
- **MODULE 02A**: Authentication
- **MODULE 02B**: Session Management
- **MODULE 03**: Cross-Site Scripting
- **MODULE 04**: Insecure Direct Object Reference
- **MODULE 05**: Security Misconfiguration
- **MODULE 06A**: Insecure Cryptographic Storage
- **MODULE 06B**: Insufficient Transport Layer Protection
- **MODULE 07**: Missing Functional-Level Access Control
- **MODULE 08**: Cross-Site Request Forgery
- **MODULE 09**: Using Known Vulnerable Components
- **MODULE 10**: Unvalidated Redirects and Forwards
Your apps are hacked, learn to defend them

DEV522: Defending Web Applications Security Essentials

This is the course to take if you have to defend web applications!

Traditional network defenses, such as firewalls, fail to secure web applications. The quantity and importance of data entrusted to web applications is growing, and defenders need to learn how to secure it. DEV522 covers the OWASP Top 10 and will help you to better understand web application vulnerabilities, thus enabling you to properly defend your organization’s web assets.

Mitigation strategies from an infrastructure, architecture, and coding perspective will be discussed alongside real-world implementations that really work. The testing aspect of vulnerabilities will also be covered so you can ensure your application is tested for the vulnerabilities discussed in class.

To maximize the benefit for a wider range of audiences, the discussions in this course will be programming-language agnostic. Focus will be maintained on security strategies rather than coding-level implementation.

DEV522: Defending Web Applications Security Essentials is intended for anyone tasked with implementing, managing, or protecting Web applications. It is particularly well suited to application security analysts, developers, application architects, pen testers, and auditors who are interested in recommending proper mitigations to web security issues, and to infrastructure security professionals who have an interest in better defending their web applications.

The course will cover the topics outlined by OWASP’s Top 10 risks document as well as additional issues the authors found of importance in their day-to-day web application development practice. The topics that will be covered include:

- Infrastructure security
- Server configuration
- Authentication mechanisms
- Application language configuration
- Application coding errors like SQL Injection and cross-site scripting
- Cross-site request forging
- Authentication bypass
- Web services and related flaws
- Web 2.0 and its use of web services
- XPATH and XQUERY languages and injection
- Business logic flaws
- Protective HTTP headers

The course will make heavy use of hands-on exercises. It will conclude with a large defensive exercise, reinforcing the lessons learned throughout the week.

Who Should Attend
- Application developers
- Application security analysts or managers
- Application architects
- Penetration testers who are interested in learning about defensive strategies
- Security professionals who are interested in learning about web application security
- Auditors who need to understand defensive mechanisms in web applications
- Employees of PCI-compliant organizations who need to be trained to comply with PCI requirements

“’What you don’t know about web app defense is most likely killing you and you wouldn’t know it.’”
-Michael Malarkey, Bank of America

Who Should Attend
- Application developers
- Application security analysts or managers
- Application architects
- Penetration testers who are interested in learning about defensive strategies
- Security professionals who are interested in learning about web application security
- Auditors who need to understand defensive mechanisms in web applications
- Employees of PCI-compliant organizations who need to be trained to comply with PCI requirements

“’What you don’t know about web app defense is most likely killing you and you wouldn’t know it.’”
-Michael Malarkey, Bank of America
Course Day Descriptions

**522.1 HANDS ON: Web Basics and Authentication Security**

We begin day one with an overview of the recent web application attack trends and and security. We follow that up with an overview of the essential technologies that are at play in web applications. You can’t win the battle if you don’t understand what you are trying to defend. We arm you with the right information so you can understand how web applications work and the security concepts related to them.

Topics: HTTP Basics; Overview of Web Technologies; Web Application Architecture; Recent Attack Trends; Authentication Vulnerabilities and Defense; Authorization Vulnerabilities and Defense

**522.2 HANDS ON: Web Application Common Vulnerabilities and Mitigations**

Since the Internet does not guarantee secrecy of information being transferred, encryption is commonly used to protect the integrity and secrecy of information on the web. We cover the security of data in transit or on disk and how encryption can help with securing that information in the context of web application security.

Topics: SSL Vulnerabilities and Testing; Proper Encryption Use in Web Application; Session Vulnerabilities and Testing; Cross-Site Request Forgery; Business Logic Flaws; Concurrency; Input-Related Flaws and Related Defense; SQL Injection Vulnerabilities, Testing and Defense

**522.3 HANDS ON: Proactive Defense and Operation Security**

Day three begins with a detailed discussion on cross-site scripting, related mitigation, and testing strategy, as well as HTTP response splitting. The code in an application may be totally locked down; however, if the server setting is insecure, the server running the application can be easily compromised. Locking down the web environment is an essential topic for discussion, so this basic concept of defending the platform and host is covered. To enable any detection of intrusion, logging and error handling must be done correctly. We will discuss the correct approach to handling incidents and handling logs. We even dive further to cover the intrusion detection aspect of web application security. In the afternoon we turn our focus to the proactive defense mechanism so that we are ahead of the bad guys in the game of hack and defend.

Topics: Cross-Site Scripting Vulnerability and Defenses; Web Environment Configuration Security; Intrusion Detection in Web Application Incident Handling; Honeytoken

**522.4 HANDS ON: AJAX and Web Services Security**

Day four of the course is dedicated to AJAX and web services security. Asynchronous JavaScript and XML (AJAX) and web services are currently the most active areas in web application development. Security issues continue to arise as organizations are diving head first into insecurely implementing new web technologies without first understanding them.

Topics: Web Services Overview; Security in Parsing of XML; XML Security; AJAX Technologies Overview; AJAX Attack Trends and Common Attacks; AJAX Defense

**522.5 HANDS ON: Cutting-Edge Web Security**

Day five has a strong focus of cutting-edge web application technologies and current research areas. Topics such as Clickjacking and DNS rebinding are covered. These vulnerabilities are difficult to defend against and require multiple defense strategies to be successful. Another topic of discussion is the new generation of single sign-on solutions such as OpenID. We cover the implication of using these authentication systems and the common gotchas to avoid.

Topics: Clickjacking; DNS Rebinding; Flash Security; Java Applet Security; Single Signon Solution and Security; IPv6 Impact on Web Security

**522.6 HANDS ON: Capture and Defend the Flag Exercise**

Day six starts with an introduction to the secure software development life cycle and how to apply it to web development. But the major focus of day six is a large lab. This lab will tie the lessons learned during the week together and reinforce the lessons by practicing them hands on. The student is provided with a virtual machine implementing a complete database-driven dynamic website. In addition, a custom tool is used to enumerate security vulnerabilities and simulate a vulnerability assessment of the website. It will be up to the student to decide which vulnerabilities are real and which are false positives. The student is then asked to mitigate the vulnerabilities. The scanner will score the student as vulnerabilities are eliminated or checked off as false positives. Advanced students will be able to extend this exercise and find vulnerabilities not presented by the scanner.

Topics: Mitigation of Server Configuration Errors; Discovering and Mitigating Coding Problems; Testing Business Logic Issues and Fixing Problems; Web Services Testing and Security Problem Mitigation

For course updates, prerequisites, special notes, or laptop requirements, visit sans.org/courses

5
Take this course to learn how to build secure Java applications and gain the knowledge and skills to:

- Keep your website from getting hacked
- Avoid becoming the next headline
- Counter a wide range of application attacks
- Prevent critical security vulnerabilities that can lead to data loss
- Understand the attackers’ mindset and how your applications can be hacked

This course teaches you the art of modern web defense for Java applications by focusing on foundational defensive techniques, cutting-edge protections, and Java EE security features that you can use in your applications as soon as you return to work. This includes learning how to:

- Identify security defects in your code
- Fix security bugs using secure coding techniques
- Utilize secure HTTP headers to prevent attacks
- Secure your sensitive REST services
- Incorporate security into your development process
- Use freely available security tools to test your applications

Great developers have traditionally distinguished themselves by the elegance, effectiveness, and reliability of their code. That’s still true, but elegance, effectiveness, and reliability have now been joined by security. This unique SANS course allows you to bone up on the skills and knowledge required to prevent your applications from getting hacked.

**How the course works**

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, hands-on 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 Java 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, 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 Java application processes cardholder data and you are required to meet PCI compliance then this course is for you.

---

**Who Should Attend**

- Developers who want to build more secure apps
- Java EE programmers
- Software engineers
- Software architects
- Application security auditors
- Technical project managers
- Senior software QA specialists
- Penetration testers who want a deeper understanding of target applications or who want to provide more detailed vulnerability remediation options

“Actually coding the examples from a find-the-weakness and fix-it standpoint, as you do in DEV541, is a big help.”

- Andrew Whitehead, Federal Reserve Bank – Richmond
Improper data validation is the root cause of the most prevalent web application vulnerabilities today. You will learn about some of the most prevalent web application vulnerabilities such as XSS, CSRF, SQL injection, HTTP response splitting, 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 Java code. The course is full of hands-on exercises where you can apply practical data validation techniques to prevent common attacks with defense ranging from input validation to output encoding and the use of new techniques like Content Security Policy.

### Topics:
- Web Application Attacks
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- SQL Injection
- HTTP Response Splitting
- Parameter Manipulation
- Directory Traversal
- Web Application Proxies
- Validation Concerns
- Character Encoding
- Input Validation
- Output Encoding
- Blacklisting & Whitelisting
- Validation Techniques
- Regular Expressions
- Servlet Filters
- Output Encoding
- Content Security Policy
- Prepared Statements
- CSRF Defense

---

### 541.2 HANDS ON: Authentication and Session Management

Broken authentication and session management are common issues that can compromise the integrity of your system. Weak authentication protections can allow an attacker to expose your most sensitive secrets—your data! You will learn about these vulnerabilities and what you can do to design and code stronger authentication protections from the start. You will learn how to use Java EE Container-Based Authentication and set up basic, form-based, and client certificate authentication. You will also learn how to protect data in transit using SSL and how to securely store passwords at rest. Various authorization attacks will be discussed as well as unvalidated forwards and redirects. Session management attacks and defenses are covered in addition to Clickjacking and associated defenses.

### Topics:
- Authentication Factors
- Authentication Attacks
- Java EE Authentication
- Basic Authentication
- Form-Based Authentication
- Client Certificates
- SSL Secure Password Storage
- Authorization
- Web and EJB Access Control
- Authorization Attacks
- Access Control Bypass
- Unvalidated Forwards and redirects
- State Management Attacks
- Session hijacking
- Session Fixation
- Clickjacking
- Using X-Frame-Options

---

### 541.3 HANDS ON: Java Platform and API Security

Java is the language of choice for the development of many mission-critical applications. As such, it is vital to understand the security features and implications of using the Java language itself and the Java Runtime Environment (JRE). Through numerous hands-on exercises you will learn about the Security Manager, how code privileges are managed, and how to sign jar files. You will also learn about Exception handling and try/catch/finally blocks as well as the importance of logging. With hands-on exercises you will also write code to encrypt both data in transit and data at rest using the Java Secure Socket Extension (JSSE) and the Java Cryptography Architecture (JCA) as well as String immutability, integer and double overflows, and learn about numerous Java language features that you should consider while writing secure code.

### Topics:
- Java Security Manager
- Permissions
- Policy File
- Jar Signing
- Error Handling
- Exceptions
- Using Try/Catch/Finally
- Logging
- Logging frameworks
- ESAPI Logging
- Encryption
- Java Secure Sockets Extension (JSSE)
- Java Cryptography Architecture (JCA)
- Class Security
- Accessibility Modifiers
- Strings
- Immutability
- String Handling
- Integer and Double Overflows
- Race Conditions
- Synchronization
- Collections
- Singletons

---

### 541.4 HANDS ON: Secure Development Lifecycle

Using what you have learned about Web application vulnerabilities, you will conduct a security review of a real-world open-source application. You will see first-hand how to integrate security in your software development life cycle (SDLC) by first conducting a code review of a large, widely used open-source application. Once you have identified various vulnerabilities in the code itself you will then perform security testing and actually exploit these weaknesses. Once they have been exploited you will then fix them using the secure coding techniques you have learned in class. The Secure Development Challenge introduces you to what is needed in a Secure SDLC and shows you how to do it.

### Topics:
- Security and the SDLC
- Conducting a Secure Code Review
- Manual Code Review
- Using a Static Analysis Tool
- Using FindBugs
- Integrating Code Review into the SDLC
- Security Testing
- Exploiting XSS, CSRF, and SQL Injection
- Secure Coding
- Fixing Weaknesses in a Running Application

For course updates, prerequisites, special notes, or laptop requirements, visit sans.org/courses
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 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 toward improving the security of .NET applications.

Rather than teaching students to use a set of tools, we’re teaching them 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 the 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, 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.
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 application 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 to prevent common attacks with defense ranging from input validation to output encoding and the 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 hands-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, and 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. 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 Nursing Society

For course updates, prerequisites, special notes, or laptop requirements, visit sans.org/courses
The C and C++ programming languages are the bedrock for most operating systems, major network services, embedded systems and system utilities. Even though C and, to a lesser extent, C++ are well understood languages, the flexibility of the language and inconsistencies in the standard C libraries have led to an enormous number of discovered vulnerabilities over the years. The unfortunate truth is that there are probably more undiscovered vulnerabilities than there are known vulnerabilities!

This course will cover all of the most common programming flaws that affect C and C++ code. The course will specifically cover the issues identified by the GSSP (GIAC Secure Software Programmer) blueprint for C/C++ with some additional items from the CERT Secure Coding Standard. Each issue is described clearly with examples. Throughout the course students are asked to identify flaws in modern versions of common open-source software to provide hands-on experience identifying these issues in existing code. Exercises also require students to provide secure solutions to coding problems in order to demonstrate mastery of the subject.

What You Will Learn

- Off-by-one errors
- Problems with Null Terminated Byte Strings
- Causes of buffer overflows
- Causes of heap overflows
- Memory management errors
- Integer promotion standards
- Side effects of integer promotions
- Common integer errors
- Common semaphore issues
- File I/O errors
- Review process for identifying coding errors
- Preventing race conditions
- Dynamic analysis tools
- Static analysis tools
- Practical defensive coding strategies

Who Should Attend

- C Programmers
- C++ Programmers
- Project Managers overseeing coding tasks in C or C++
- Embedded programmers working with C or C++
- Legacy code maintainers
- Code auditors

SANS has done a great job over the years of assisting the industry in performing triage. We’ve progressed from needing to secure our perimeters to giving advice on how to monitor networks and identify attacks, how to deploy services securely and how to secure operating systems. Now that the triage is done it’s time for us to get to the heart of our real problems: we’ve got a lot of bad code that we’re relying on for mission-critical applications. This course adds one more tool to your arsenal, allowing you to identify and fix your problems at the source… literally!

-David Hoelzer
Prerequisites

Students should have at least several months of coding experience, preferably web application coding experience. It is best if the student is familiar with one of the following languages: Perl, PHP, C, C++, Java or Ruby.

DEVELOPER 536
Secure Coding: Developing Defensible Applications

Two-Day Program
12 CPEs
Laptop Required
DELIVERY METHODS:
▶ Onsite

The audit procedure documents for PCI 1.2 tell auditors that they should look for evidence that web application programmers in a PCI environment have had “training for secure coding techniques.” The problem that many businesses are facing, however, is, “What is that training and where can I get it?” This course packs a thorough explanation and examination of the OWASP Top 10 issues, which are the foundation of the PCI requirement, into a two-day course.

Throughout the course we will look at examples of the types of flaws that secure coding protects against, examine how the flaw might be exploited and then focus on how to correct that code. Coupled with the lectures, there are 10 hands-on exercises where the students will have the opportunity to test out their new skills identifying flaws in code, fixing code and writing secure code. All of the exercises are available in Perl, PHP, C/C++, Ruby and Java. This will allow the students to try their hand at any of the major web application coding languages that they work with in addition to some of the supporting languages that might be at work behind the scenes. Students are not required to be familiar with all of these languages but should be proficient in at least one of them. Lectures are presented using a more or less code-neutral format.

For more information on this course, visit the website sans.org/dev536


You Will Be Able To

- Apply a detailed, four-step methodology to your web application penetration tests, including recon, mapping, discovery, and exploitation.
- Analyze the results from automated web testing tools to remove false positives and validate findings.
- Use Python to create testing and exploitation scripts during a penetration test.
- Create configurations and test payloads within Burp Intruder to perform SQL injection, XSS, and other web attacks.
- Use FuzzDB to generate attack traffic to find flaws such as Command Injection and File Include issues.
- Assess the logic and transaction flow within a target application to find logic flaws and business vulnerabilities.
- Use Durzosploit to obfuscate XSS payloads to bypass WAFs and application filtering.
- Analyze traffic between the client and the server application using tools such as Ratproxy and Zed Attack Proxy to find security issues within the client-side application code.
- Use BeEF to hook victim browsers, attack the client software and network, and evaluate the potential impact XSS flaws have within an application.
- Perform a complete web penetration test during the Capture the Flag exercise to pull all of the techniques and tools together into a comprehensive test.

Who Should Attend

- General security practitioners
- Penetration testers
- Ethical hackers
- Web application developers
- Website designers and architects

Assess Your Web Apps in Depth

Web applications are a major point of vulnerability in organizations today. Web app holes have resulted in the theft of millions of credit cards, major financial and reputational damage for hundreds of enterprises, and even the compromise of thousands of browsing machines that visited websites altered by attackers. In this intermediate to advanced level class, you’ll learn the art of exploiting web applications so you can find flaws in your enterprise’s web apps before the bad guys do. Through detailed, hands-on exercises and training from a seasoned professional, you will be taught the four-step process for web application penetration testing. You will inject SQL into backend databases, learning how attackers exfiltrate sensitive data. You will utilize cross-site scripting attacks to dominate a target infrastructure in our unique hands-on laboratory environment. And you will explore various other web app vulnerabilities in depth with tried-and-true techniques for finding them using a structured testing regimen. You will learn the tools and methods of the attacker so that you can be a powerful defender.

Throughout the class, you will learn the context behind the attacks so that you intuitively understand the real-life applications of our exploitation. In the end, you will be able to assess your own organization’s web applications to find some of the most common and damaging web application vulnerabilities today.

By knowing your enemy, you can defeat your enemy. General security practitioners, as well as website designers, architects, and developers, will benefit from learning the practical art of web application penetration testing in this class.

“Fun while you learn! Just don’t tell your manager. Every class gives you invaluable information from real-world testing you cannot find in a book.”

-David Fava, The Boeing Company
542.1 HANDS ON: The Attacker's View of the Web

We begin by examining web technology — protocols, languages, clients, and server architectures — from the attacker's perspective. Then we cover the four steps of web application pen tests: reconnaissance, mapping, discovery, and exploitation.

Topics: Overview of the Web from a Penetration Tester's Perspective; Exploring the Various Servers and Clients; Discussion of the Various Web Architectures; Discover How Session State Works; Discussion of the Different Types of Vulnerabilities; Define a Web Application Test Scope and Process; Define Types of Penetration Testing

542.2 HANDS ON: Reconnaissance and Mapping

Reconnaissance includes gathering publicly-available information regarding the target application and organization, identifying machines that support our target application, and building a profile of each server. In this section, we will build a map of the application by identifying the components, analyzing the relationship between them, and determining how they work together.

Topics: Discover the Infrastructure Within the Application; Identify the Machines and Operating Systems; SSL Configurations and Weaknesses; Explore Virtual Hosting and its Impact on Testing; Learn Methods to Identify Load Balancers; Software Configuration Discovery; Explore External Information Sources; Google Hacking; Learn Tools to Spider a Website; Scripting to Automate Web Requests and Spidering; Application Flow Charting; Relationship Analysis Within an Application; JavaScript for the Attacker

542.3 HANDS ON: Server-Side Discovery

We will continue with the discovery phase, exploring both manual and automated methods of discovering vulnerabilities within the applications as well as exploring the interactions between the various vulnerabilities and the different user interfaces that web apps expose to clients.

Topics: Learn Methods to Discover Various Vulnerabilities; Explore Differences Between Different Data Back-ends; Explore Fuzzing and Various Fuzzing Tools; Discuss the Different Interfaces Websites Contain; Understand Methods for Attacking Web Services

542.4 HANDS ON: Client-Side Discovery

Learning how to discover vulnerabilities within client-side code, such as Java applets and Flash objects, including use of tools to decompile the objects and applets. We will have a detailed discussion of how AJAX and web service technology enlarges the attack surface that pen testers leverage.

Topics: Learn Methods to Discover Various Vulnerabilities; Learn Methods to Decompile Client-side Code; Explore Malicious Applets and Objects; Discovery Vulnerabilities in Web Application Through Their Client Components; Understand Methods for Attacking Web Services; Understand Methods for Testing Web 2.0 and AJAX-based Sites; Learn How AJAX and Web Services Change Penetration Tests; Learn the Attacker's Perspective on Python and PHP

542.5 HANDS ON: Exploitation

Launching exploits against real-world applications includes exploring how they can help in the testing process, gaining access to the browser history, port scanning internal networks, and searching for other vulnerable web applications through zombie browsers.

Topics: Explore Methods to Zombify Browsers; Discuss Using Zombies to Port Scan or Attack Internal Networks; Explore Attack Frameworks; Walk Through an Entire Attack Scenario; Exploit the Various Vulnerabilities Discovered; Leverage the Attacks to Gain Access to the System; Learn How to Pivot our Attacks Through a Web Application; Understand Methods of Interacting with a Server Through SQL Injection; Exploit Applications to Steal Cookies; Execute Commands Through Web Application Vulnerabilities

542.6 HANDS ON: Capture the Flag

The goal of this event is for students to use the techniques, tools, and methodology learned in class against a realistic intranet application. Students will be able to use a virtual machine with the SamuraiWTF web pen testing environment in class and can apply that experience in their workplace.

Topics: Capture the Flag

For course updates, prerequisites, special notes, or laptop requirements, visit sans.org/courses
SECURITY 642

Advanced Web App Penetration Testing and Ethical Hacking

Six-Day Program
36 CPEs
Laptop Required

DELIVERY METHODS:
Live Events
OnDemand
SelfStudy
Onsite

This course is designed to teach you the advanced skills and techniques required to test web applications. This advanced pen testing course uses a combination of lecture, real-world experiences, and hands-on exercises to educate you in the techniques used to test the security of enterprise applications. The final day of the course culminates in a Capture the Flag (CtF) event that tests the knowledge you will have acquired the previous five days.

We will begin by exploring specific techniques and attacks to which applications are vulnerable. These techniques and attacks use advanced ideas and skills to exploit the system through various controls and protections. This learning will be accomplished through lectures and exercises using real-world applications.

We will then explore encryption as it relates to web applications. You will learn how encryption works as well as techniques to identify the type of encryption in use within the application. Additionally, you will learn methods for exploiting or abusing this encryption, again through lecture and labs.

The next day of class will focus on how to identify web application firewalls, filtering, and other protection techniques. You will then learn methods to bypass these controls in order to exploit the system. You’ll also gain skills in exploiting the control itself to further the evaluation of the security within the application.

Following these general exploits, you will learn techniques that target specific enterprise applications. You will attack systems such as content management and ticketing systems. We will explore the risks and flaws found within these systems and how to better exploit them. This part of the course will also include web services and mobile applications, as these are prevalent within modern organizations.

This information-packed advanced pen testing course will wrap up with a full-day Capture the Flag that will target an imaginary organization’s web applications and include both Internet and intranet applications of various technologies. This event is designed to allow you to put the pieces together from the previous five days, reinforcing the information and learning you will have gained.

Who Should Attend
- Web penetration testers
- Security consultants
- Developers
- QA testers
- System administrators
- IT managers
- System architects

You Will Be Able To
- Assess and attack complex modern applications
- Understand the special testing and exploits available against content management systems such as SharePoint and WordPress
- Use techniques to identify and attack encryption within applications
- Identify and bypass web application firewalls and application filtering techniques to exploit the system
- Use exploitation techniques learned in class to perform advanced attacks against web application flaws such as XSS, SQL injection and CSRF

“Outstanding course!! It is great to have an opportunity to learn the material from someone who is extremely relevant in the field and is able to impart the value of his experiences.”

-Bobby Bryant, DoD
Course Day Descriptions

642.1 HANDS ON: Advanced Discovery and Exploitation

As applications and their vulnerabilities become more complex, penetration testers have to be able to handle these targets. We will begin the class by exploring how Burp Suite works and more advanced ways to use it within your penetration-testing processes. The exploration of Burp Suite will focus on its ability to work within the traditional web penetration testing methodology and assist in manually discovering the flaws within the target applications. Following this discussion, we will move into studying specific vulnerability types. This examination will explore some of the more advanced techniques for finding server-based flaws such as SQL injection. After discovering the flaws, we will then work through various ways to exploit these flaws beyond the typical means exhibited today. These advanced techniques will help penetration testers show the risks to which the flaws expose an organization.

Topics: Review of the Testing Methodology; Using Burp Suite in a Web Penetration Test; Examining How to Use Burp Intruder to Effectively Fuzz Requests; Exploring Advanced Discovery Techniques for SQL Injection and Other Server-Based Flaws; Learning Advanced Exploitation Techniques

642.2 HANDS ON: Discovery and Exploitation for Specific Applications

We will continue the exploration of advanced discovery and exploitation techniques for today’s complex web applications. We’ll start by exploring advanced client-side flaws such as combined cross-site scripting (XSS) and cross-site request forgery (XSRF) vulnerabilities. We will explore some of the more advanced methods for discovering these issues. After finding the flaws, you will learn some of the more advanced methods of exploitation, such as scriptless attacks and building web-based worms using XSRF and XSS flaws within an application. During the next part of the day we’ll explore various popular applications and frameworks and how they change the discovery techniques within a web penetration test. This section of the class examines applications such as SharePoint and WordPress. These specific targets have unique needs and features that make testing them both more complex and more fruitful for the tester. This section of the class will help you understand these differences and make use of them in your testing.

Topics: Discovering XSRF Flaws Within Complex Applications; Learning About DOM-based XSS Flaws and How to Find Them Within Applications; Exploiting XSS Using Scriptless Injections; Bypassing Anti-XSRF Controls Using XSS/XSRF Worms; Attacking SharePoint Installations; How to Modify Your Test Based on the Target Application

642.3 HANDS ON: Web Application Encryption

Cryptography weaknesses are a common area where flaws are present, yet few penetration testers have the skill to investigate, attack and exploit these flaws. When we investigate web application crypto attacks, we typically target the implementation and use of cryptography in modern web applications. Many popular web programming languages or development frameworks make encryption services available to the developer, but do not inherently protect encrypted data from being attacked, or permit the developer to use cryptography in a weak manner. These implementation mistakes are going to be our focus in this section, as opposed to the exploitation of deficiencies in the cryptographic algorithms themselves. We will also explore the various ways applications use encryption and hashing insecurely. Students will learn techniques such as identifying what the encryption technique is and how to exploit various flaws within the encryption or hashing.

Topics: Exploring How to Identify the Cryptography in Use; Discovering How to Attack the Encryption Keys; Learning How to Attack Electronic Codebook (ECB) Mode Ciphers; Exploiting Padding Oracles and Cipher Block Chaining (CBC) Bit Flipping

642.4 HANDS ON: Mobile Applications and Web Services

Web applications are no longer limited to the traditional HTML-based interface. Web services and mobile applications have become more common and are regularly being used to attack clients and organizations. As such, it has become very important that penetration testers understand how to evaluate the security of these systems. After finishing our discussion on cryptography attacks, you will learn how to build a test environment for testing web services used by mobile applications. We will also explore various techniques to discover flaws within the applications and backend systems. These techniques will make use of tools such as Burp Suite and other automated toolsets.

Topics: Attacking CBC Chosen Plaintext; Exploiting CBC with Padding Oracles; Understanding the Mobile Platforms and Architectures; Intercepting Traffic to Web Services and from Mobile Applications; Building a Test Environment; Penetration Testing of Web Services

642.5 HANDS ON: Web Application Firewall and Filter Bypass

Today, applications are using more security controls to help prevent attacks. These controls, such as Web Application Firewalls and filtering techniques, make it more difficult for penetration testers during their testing. These controls block many of the automated tools and simple techniques used to discover flaws. On day 5 you will explore techniques used to map the control and how it is configured to block attacks. You’ll be able to map out the rule sets and determine the specifics of how they detect attacks. This mapping will then be used to determine attacks that will bypass the control. You’ll use HTML5, UNICODE and other encodings that will enable your discovery techniques to work within the protected application.

Topics: Understanding of Web Application Firewalls and Filtering Techniques; Exploring How to Determine the Rule Sets Protecting the Application; Learning How HTML5 Injections Work; Discovering the Use of UNICODE and Other Encodings

642.6 HANDS ON: Capture the Flag

During day six of the class, you will be placed on a network and given the opportunity to complete an entire penetration test. The goal of this Capture the Flag event is for you to explore the techniques, tools, and methodology you will have learned over the last five days. You’ll be able to use these ideas and methods against a realistic extranet and intranet. At the end of the day, you will provide a verbal report of the findings and methodology you followed to complete the test. Students will be provided with a virtual machine that contains the Samurai Web Testing Framework web penetration-testing environment. You will be able to use this both in the class and when you return to your jobs.

For course updates, prerequisites, special notes, or laptop requirements, visit sans.org/courses
Mobile phones and tablets have become essential to enterprise and government networks, from small organizations to Fortune 500 companies and large-scale agencies. Often, mobile phone deployments grow organically, adopted by multitudes of end-users for convenient email access as well as by managers and executives who need access to sensitive organizational resources from their favored personal mobile devices. In other cases, mobile phones and tablets have become critical systems for a wide variety of production applications from ERP to project management. With increased reliance on these devices, organizations are quickly recognizing that mobile phones and tablets need greater security implementations than a simple screen protector and clever password.

Whether the device is an Apple iPhone or iPad, a Windows Phone, an Android or a BlackBerry phone or tablet, the ubiquitous mobile device has become a hugely attractive and vulnerable target for nefarious attackers. The use of mobile devices introduces a vast array of new risks to organizations, including:

- Distributed sensitive data storage and access mechanisms
- Lack of consistent patch management and firmware updates
- The high probability of device loss or theft, and more.

Mobile code and apps are also introducing new avenues for malware and data leakage, exposing critical enterprise secrets, intellectual property, and personally identifiable information assets to attackers. To further complicate matters, today there simply are not enough people with the security skills needed to manage mobile phone and tablet deployments.

This course was designed to help organizations struggling with mobile device security by equipping personnel with the skills needed to design, deploy, operate, and assess a well-managed secure mobile environment. From practical policy development to network architecture design and deployment, and from mobile code analysis to penetration testing and ethical hacking, this course will help you build the critical skills necessary to support the secure deployment and use of mobile phones and tablets in your organization.

You will gain hands-on experience in designing a secure mobile phone network for local and remote users and learn how to make critical decisions to support devices effectively and securely. You will also be able to analyze and evaluate mobile software threats, and learn how attackers exploit mobile phone weaknesses so you can test the security of your own deployment. With these skills, you will be a valued mobile device security analyst, fully able to guide your organization through the challenges of securely deploying mobile devices.

Who Should Attend
- Penetration testers
- Ethical hackers
- Auditors who need to build deeper technical skills
- Security personnel whose job involves assessing target networks and systems to find security vulnerabilities

"Eye-opening material. I am mesmerized by the course, SEC575. It’s time to test apps, they can’t be trusted." -Matthew Britton, BCBSLA

"SEC575 is simply eye opening. Organizations are so busy trying to roll out their BYOD projects without any understanding of the risks. This course is a must for security professionals rolling out BYOD projects." -Vijay Kora, Open Solutions Consulting Inc.
The first part of the course looks at the significant threats affecting mobile phone deployment and how organizations are being attacked through these systems. As a critical component of a secure deployment, we guide you through the process of defining mobile phone and tablet policies with sample policy language and recommendations for various vertical industries, taking into consideration the legal obligations of enterprise organizations. We’ll also look at the architecture and technology behind mobile device infrastructure systems for Apple, Android, BlackBerry, and Windows devices, as well as the platform-specific security controls available including device encryption, remote data wipe, application sandboxing, and more.

**Topics:** Mobile Phone and Tablet Problems and Opportunities; Mobile Devices and Infrastructure; Mobile Phone and Tablet Security Models; Legal aspects of Mobile; Mobile Device Policy Considerations and Development

With an understanding of the threats, architectural components and desired security methods, we can design and implement device and infrastructure systems to defend against these threats. In this part of the course, we’ll examine the design and deployment of network and system infrastructure to support a mobile phone deployment including the selection and deployment of Mobile Device Management (MDM) systems.

**Topics:** Wireless Network Infrastructure; Remote Access Systems; Certificate Deployment Systems; Mobile Device Management (MDM) System Architecture; Mobile Device Management (MDM) Selection

With the solid analysis skills taught in this section of the course, we can evaluate apps to determine the type of access and information disclosure threats that they represent. Security professionals can use these skills not only to determine which outside applications the organization should allow, but also to evaluate the security of any apps developed by the organization itself for its employees or customers. In this process, we’ll use jailbreaking and other techniques to evaluate the data stored on mobile phones.

**Topics:** Unlocking, Rooting, and Jailbreaking Mobile Devices; Mobile Phone Data Storage and Filesystem Architecture; Filesystem Application Modeling; Network Activity Monitoring; Mobile Code and Application Analysis; Approving or Disapproving Applications in Your Organization

Through ethical hacking and penetration testing, we examine the mobile devices and infrastructure from the perspective of an attacker, identifying and exploiting flaws that could allow unauthorized access to data or supporting networks. By identifying and understanding the implications of these flaws, we can evaluate the mobile phone deployment risk to the organization with practical, useful risk metrics.

**Topics:** Fingerprinting Mobile Devices; WiFi Attacks; Bluetooth Attacks; Network Exploits

Continuing our look at ethical hacking and penetration testing, we turn our focus to exploiting weaknesses on individual mobile devices including iPhones, iPads, Android phones, Windows Phones and BlackBerry phones and tablets. We’ll also examine platform-specific application weaknesses and look at the growing use of web framework attacks.

**Topics:** Mobile Device Exploits; Web Framework Attacks; Application Attacks; Cloud/Remote Data Accessibility Attacks

On the last day of class, we apply the skills, concepts, and technology covered in the course for a comprehensive Capture the Flag event. In this day-long, in-depth hands-on exercise, you will:

- Have the option to participate in multiple organizational roles related to mobile device security
- Design a secure infrastructure for the deployment of mobile phones
- Monitor network activity to identify attacks against mobile devices
- Extract sensitive data from a compromised iPad
- Attack a variety of mobile phones and related network infrastructure components.

In the exercise, you will use the skills built throughout the course to evaluate real-world systems and defend against attackers, simulating the realistic environment you’ll face when you get back to the office. You will leave the course armed with the knowledge and skills you’ll need to securely integrate and deploy mobile devices in your organization.
GIAC offers a free skills assessment where participants can test their ability to code securely in Java or .Net. Each assessment has 50 questions and a 2½ hour time limit. After taking the test, participants will receive an Assessment Report detailing the skills mastered and the areas where skills could be refined. The assessment will gauge a participant's ability to avoid the most dangerous threats identified in the OWASP Top 10 and the CWE/SANS Top 25. Below is a sample of the Assessment Report:

### Assessment Category Detail: Java

<table>
<thead>
<tr>
<th>TASK</th>
<th>RESULTS</th>
<th>RISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Validation Principles</td>
<td>63%</td>
<td></td>
</tr>
<tr>
<td>Parameterized Queries/Prepared Statements</td>
<td>63%</td>
<td></td>
</tr>
<tr>
<td>Input Validation Techniques</td>
<td>47%</td>
<td></td>
</tr>
<tr>
<td>Encryption of Data at Rest</td>
<td>48%</td>
<td></td>
</tr>
<tr>
<td>Exception Handling</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>Input Validation Sources</td>
<td>64%</td>
<td></td>
</tr>
<tr>
<td>Race Conditions</td>
<td>51%</td>
<td></td>
</tr>
<tr>
<td>Class/Package/Method Access Modifiers</td>
<td>59%</td>
<td></td>
</tr>
<tr>
<td>Output Encoding</td>
<td>57%</td>
<td></td>
</tr>
<tr>
<td>Programmatic Access Control</td>
<td>66%</td>
<td></td>
</tr>
<tr>
<td>When to Authenticate</td>
<td>61%</td>
<td></td>
</tr>
<tr>
<td>Session Protection</td>
<td>66%</td>
<td></td>
</tr>
<tr>
<td>Declarative Access Control</td>
<td>63%</td>
<td></td>
</tr>
<tr>
<td>Authentication Techniques</td>
<td>55%</td>
<td></td>
</tr>
<tr>
<td>Restricting Access to Functions</td>
<td>65%</td>
<td></td>
</tr>
<tr>
<td>Restricting Access to Resources</td>
<td>79%</td>
<td></td>
</tr>
<tr>
<td>Authentication Protection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication Responsibilities</td>
<td>62%</td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>72%</td>
<td></td>
</tr>
<tr>
<td>Class File Protection</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>Fail-Safe Connection Patterns</td>
<td>67%</td>
<td></td>
</tr>
<tr>
<td>Communications Encryption</td>
<td>55%</td>
<td></td>
</tr>
<tr>
<td>J2EE Filters</td>
<td>64%</td>
<td></td>
</tr>
<tr>
<td>Singletons &amp; Shared Resources</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>Configuration of Error Handling</td>
<td>58%</td>
<td></td>
</tr>
<tr>
<td>Class Security</td>
<td>63%</td>
<td></td>
</tr>
<tr>
<td>Garbage Collector</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>Array List vs. Vector</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Code Privileges</td>
<td>59%</td>
<td></td>
</tr>
<tr>
<td>Integer and Double Overflows</td>
<td>41%</td>
<td></td>
</tr>
<tr>
<td>java.lang.string</td>
<td>63%</td>
<td></td>
</tr>
<tr>
<td>JAAS</td>
<td>52%</td>
<td></td>
</tr>
</tbody>
</table>

**Developers in this sample:**

Average score is 60%. Minimum target is 70%.

The group demonstrated acceptable skill level on only 5 of 32 programming tasks.

**Red cells** indicate tasks that are highly critical AND where the group demonstrated significant skill gaps.

**EXAMPLE:** Errors in input validation and authentication programming techniques are two primary causes of vulnerabilities. This group answered about half of the questions correctly.

**Green cells** indicate tasks that are very important AND where the group demonstrated combined strength.

**EXAMPLE:** Exception handling, restricting access to resources and authentication protection are important topics where the group scored at or above target.
Experts Announce Agreement on the 25 Most Dangerous Software Errors – And How to Fix Them

In Washington, D.C., experts from more than 30 U.S. and international cybersecurity organizations jointly released a new list of the 25 most dangerous software errors that enable security bugs, cyber espionage, and cyber crime. These 25 software errors, and their “on the cusp cousins” have been the cause of nearly every major type of cyber attack, including recent penetrations of Google, power systems, military systems, and millions of other attacks on small businesses and home users. A global effort to eliminate these software errors is the first step against organized cyber criminals and the persistent threat from competing nation states.

In addition to the most common software errors, acquisition experts agreed on a standard for contract language between software buyers and developers. The use of this contract language helps ensure buyers are not held liable for software containing faulty code. Coding errors are a common gateway for attackers to penetrate networks.

How Will the Top 25 Software Errors Be Used?

The Top 25 Software Errors will have four major impacts:

• Software buyers will be able to buy much safer software.
• Employers will be able to ensure they have programmers who can write more secure code.
• Programmers will have tools that consistently measure the security of the software they are writing.
• Colleges will be able to teach secure coding more confidently.

How Many of These Questions Can You Answer with Confidence?

• Where are the gaps in our programmers’ secure coding knowledge and skills?
• Which of our programmers and contractors have the strongest secure coding skills?
• Do any of the current job candidates or potential contractors have solid secure programming skills?
• Do we have at least one security-savvy programmer on every critical development project?

Need help answering any of these questions?
A SANS Corporate License for training will save you money!

SANS Corporate License = Dramatically Reduced Costs

What is a Corporate License and what does it include?

• Highest-quality SANS training
• Unlimited reach
• Rapid deployment
• Measureable results
• Unlimited SANS training online
• Full course curriculum
• .mp3 audio files for offline study and mobile devices
• Future proof — any new courses already included by track
• Updates
• Dramatically-reduced costs
Secure Development Techniques

Excerpt from the WhatWorks Top 35 Secure Development Techniques – See the full list at sans.org/whatworks/poster-spring-2010.pdf

Java/JEE Tips

1) Perform data validation with a security API such as OWASP ESAPI

See the following paper for some examples that use ESAPI for data validation:
sans.org/reading_room/application_security/protecting_web_apps.pdf

2) Use PreparedStatements with properly bound variables

BAD:
String query = "SELECT id FROM users WHERE userid = '' + userid + "";  
PreparedStatement stmt = con.prepareStatement(query);  
ResultSet rs = stmt.executeQuery();

GOOD:
String query = "SELECT id FROM users WHERE userid = ?";  
PreparedStatement stmt = con.prepareStatement(query);  
stmt.setString(1, userid);  
ResultSet rs = stmt.executeQuery();

3) Don’t perform security-critical operations based on data from HttpServletRequest parameters

BAD:
String role = request.getParameter("role");  
if (role != null && role.equals("admin") {  
   // do admin stuff
}

4) Use a framework like Spring Security or ESAPI for authentication and authorization

See the following sites for additional information:
http://static.springsource.org/spring-security
http://owasp.org/index.php/ESAPI

5) Don’t use instance variables in Servlets

BAD:
public class BadServlet extends HttpServlet {  
   private String primaryKey;  // don’t do this!
   ...
}
**PHP Tips**

1) Use prepared SQL statements.

**BAD:**
```sql
mysql_db_query("select id from users where username='"$Username"'")
```

**BETTER:**
```php
 Stmt=$DB->prepare("select id from users where username=?");
 Stmt=$DB->bind_param("s",$Username);
 Stmt->execute();
```

2) Enable and configure Suhosin

See [hardened-php.net/suhosin](http://hardened-php.net/suhosin) for details about Suhosin.

3) Extract data from super globals inside validation functions only

**BAD:**
```php
 $UserID=$_POST['userid'];
 if ( ! is_int($userID) ) {
   $UserID=0;
 }
```

**BETTER:**
```php
 function get_userid($name) {
   $value=$_POST[$name];
   if ( is_int($value) ) {
     return $value;
   }
   return FALSE;
 }
```

4) Replace “print” statements with a wrapper function escaping HTML tags like

**BAD:**
```php
 print $value;
```

**BETTER:**
```php
 function safe_out($value) {
   $value=htmlentities($value,ENT_QUOTES,'UTF-8');
   print $value;
 }
```

---

**C and C++ Tips**

1) Validate input from all untrusted data sources.

2) Compile code using the highest warning level available for your compiler and eliminate warnings by modifying the code.

3) Create a software architecture and design your software to implement and enforce security policies.

4) Keep the design as simple and small as possible.

5) Base access decisions on permission rather than exclusion.
.NET Tips

1) For data validation, follow the Constrain, Reject/Replace, Assign (to local variable) paradigm.

2) Use a validation abstraction layer to make validating data easier and more consistent.

3) Validate data from any and all untrusted sources – including cookies, URL parameters, Form Fields, and HTTP Headers, as well as inputs from external systems.

Code example combined for first three items above:

```csharp
string sanitizedLastName = null;
ValidationUtility.TryValidateAndSanitizeLastName(txtLastName.Text, out sanitizedLastName) {
    // Success, use sanitizedLastName. Never use txtLastName.Text
    // again. Simplifies code review.
} else {
    // Failed, NEVER display txtLastName.Text back to user or use
    // again in code
}
```

// Centralize Validation

```csharp
public class ValidationUtility {
    public static bool TryValidateAndSanitizeLastName(string unsanitizedLastName, out string sanitizedLastName) {
        bool isValid = false;
        // Step 1: Constrain. Use whitelists, not blacklists.
        if (Regex.IsMatch(unsanitizedLastName, "^[a-z']+$", RegexOptions.IgnoreCase)) {
            // Step 2: Replace, substitute any potential bad characters with
            // something safe for storage. E.g., the tick ‘ char with the
            // pipe | char
            unsanitizedLastName = unsanitizedLastName.Replace('\', '|');
            isValid = true;
            // 3. Assign
            sanitizedLastName = unsanitizedLastName;
        } else {
            // Communicate intent to humans reading the code.
            isValid = false;
            sanitizedLastName = null;
        }
        return isValid;
    }
}
```

4) Use Microsoft’s AntiXSS library to counter XSS attacks.

Encode all untrusted output.


```csharp
<div>Welcome, <%= AntiXss.HtmlEncode(Request.Form["FullName"]); %></div>
```
Stay In Touch with the AppSec Community

Find free resources and materials related to application security. SANS Software Security Institute website features the App Sec Street Fighter blog, free research, news, and resources to keep you up to date with the most recent attack vectors and application vulnerabilities, as well as full course descriptions of the developer curriculum, information on GIAC certification, and upcoming events. New content is added regularly, so please visit often. And don’t forget to share this information with your fellow developers and security professionals.

software-security.sans.org

1. Top 25 Software Errors
Learn about the 25 most dangerous software errors that enable security bugs, cyber espionage, and cyber crime. See page 9 for more information. sans.org/top25-programming-errors

2. Procurement language
Draft language to help you ensure that contracts for application development, management and maintenance require the contractors to consider and build security into the process. This concept has proven critical to countless organizations by helping them avoid security issues late in the development cycle and by eliminating or significantly mitigating potential threats resulting from insecure code.
software-security.sans.org/appseccontract

3. Application Security Street Fighter Blog
software-security.sans.org/blog

4. AppSec Webcasts
software-security.sans.org/resources/webcasts

5. Software Security Poster
software-security.sans.org/free-poster

6. Twitter
@SANSAppSec

7. Assessments to gauge developers’ and contractors’ abilities to code securely in a specific language: Java and .NET
Contact spa@sans.org for more information.
Frank Kim – SANS Certified Instructor and Curriculum Lead for SANS’ Software Security Curriculum

Frank Kim is a security leader with over 16 years of experience in information security, risk management, and enterprise IT. He has a passion for developing security strategies and building teams focused on practical solutions to business risks. He currently serves as the curriculum lead for application security at the SANS Institute and is the author and an instructor for the Secure Coding in Java course. Frank has presented at security, leadership, and software development events around the world including JavaOne, Devoxx, Jazoon, and UberConf and is a two-time JavaOne Rock Star. @sansappsec

Johannes Ullrich, PhD – SANS Senior Instructor

As Dean of Research for the SANS Technology Institute, Johannes is currently responsible for the SANS Internet Storm Center (ISC) and the GIAC Gold program. In 2000, he founded DShield.org, which is now the data collection engine behind the ISC. His work with the ISC has been widely recognized, and in 2004, Network World named him one of the 50 most powerful people in the networking industry. Prior to working for SANS, Johannes worked as a lead support engineer for a web development company and as a research physicist. Johannes holds a PhD in Physics from SUNY Albany and is located in Jacksonville, Florida. His daily podcast summarizes current security news in a concise format. @johullrich @sans_isc

Justin Searle – SANS Senior Instructor

Justin Searle is a managing partner of UtiliSec, specializing in Smart Grid security architecture design and penetration testing. Justin led the Smart Grid Security Architecture group in the creation of NIST Interagency Report 7628 and played key roles in the Advanced Security Acceleration Project for the Smart Grid (ASAP-SG). He currently leads the testing group at the National Electric Sector Cybersecurity Organization Resources (NESCOR). Justin has taught courses in hacking techniques, forensics, networking, and intrusion detection for multiple universities, corporations, and security conferences. In addition to electric power industry conferences, Justin frequently presents at top international security conferences such as Black Hat, DEFCON, OWASP, Nullcon, and AusCERT. Justin co-leads prominent open-source projects including the Samurai Web Testing Framework (SamuraiWTF), the Samurai Security Testing Framework for Utilities (SamuraiSTFU), Middler, Yokosol, and Laudanum. Justin has an MBA in International Technology and is a CISSP and SANS GIAC certified Incident Handler (GCIH), Intrusion Analyst (GCIA), and Web Application Penetration Tester (GWAPT). @meeas

Seth Misenar – SANS Principal Instructor

Seth Misenar serves as lead consultant and founder of Jackson, Mississippi-based Context Security, which provides information security though leadership, independent research, and security training. Seth’s background includes network and web application penetration testing, vulnerability assessment, regulatory compliance efforts, security architecture design, and general security consulting. He has previously served as both physical and network security consultant for Fortune 100 companies as well as the HIPAA and information security officer for a state government agency. Prior to becoming a security geek, Seth received a BS in philosophy from Millsaps College, where he was twice selected for a Ford Teaching Fellowship. Also, Seth is no stranger to certifications and thus far has achieved credentials which include, but are not limited to, the following: CISSP, GPEN, GWAPT, GSEC, GCIA, GCIH, GCWN, GCFA, and MCSE. @sethmisenar
Eric Johnson – SANS Instructor & SANS Curriculum Product Manager

Eric Johnson is a security consultant at Cypress Data Defense, and an instructor and contributing author for the SANS DEV544 Secure Coding in .NET course. He previously spent six years performing web application security assessments for a large financial institution, and another four years focusing on ASP .NET web development. Other experience includes developing security tools, secure code review, vulnerability assessment, penetration testing, risk assessment, static source code analysis, and security research. Eric completed a BS in computer engineering and a master of science in information assurance at Iowa State University. Eric currently holds the GSSP-.NET, GWAPT, and CISSP certifications and is located in West Des Moines, Iowa.

Megan Restuccia – SANS Certified Instructor

Megan is currently a certified instructor with the SANS Institute as well as a senior engineer with Savvis. She has over 16 years of experience in information technology with an extensive background in secure application infrastructure design/management utilizing Linux and Windows environments for both small and large implementations. Her experience spans several verticals, including financial services, healthcare, education, and telecommunications, allowing her to have a well-rounded understanding of various business needs. Megan holds several professional certifications from Red Hat, Cisco, ISC2, and SANS. She also holds a BS in computer science and an MBA from Columbia University. Megan’s most recent focuses were on DLP, security regulations, secure applications design and training, secure infrastructure design, and vendor risk assessments.

Jason Lam – SANS Certified Instructor

Jason is a senior security analyst at a major financial institution in Canada. His recent SANS Institute courseware development includes Defending Web Application Security Essentials and Web Application Pen Testing Hands-On Immersion. Jason started his career as a programmer before moving on to ISP network administration, where he handled network security incidents, which sparked his interest in information security. Jason specializes in web application security, penetration testing, and intrusion detection. He currently holds a BA in Computer Science from York University in Toronto, Ontario, as well as the CISSP, GCIA, GCFW, GCUX, GCWN, and GCIH certifications. @jasonlam_sec

Gregory Leonard – SANS Instructor

Gregory Leonard has over 16 years of experience in software development, with an emphasis on writing large-scale enterprise applications. Greg’s responsibilities have included application architecture and security, performing infrastructure design and implementation, security analysis, code reviews, and evaluating performance diagnostics. Greg is currently focusing on overseeing the integration of secure development practices for his company.

Mano Paul – SANS Instructor

Mano Paul is a seasoned veteran in the discipline of information security, software assurance and software development, spanning responsibilities that include designing and developing security programs from compliance to coding, security in the software development lifecycle, risk management, security strategy, awareness, training and education. He is the CEO of SecuRisk Solutions and Express Certifications, companies that specialize in information security training, product development, consulting, and certification assessment. He is also (ISC)² appointed software assurance advisor and a member of the Application Security Advisory Council. He holds the CISSP, CSSLP, GWAPT, GSSP-.Net, MCSD, MCAD, ECSA and CompTIA Network+ certifications. @manopaul
GIAC Certified Secure Software Programmers (GSSP) have the knowledge, skills, and abilities to write secure code and recognize security shortcomings in existing code. GIAC certifications allow individuals to prove they are trained and qualified to work on the development and maintenance of your critical applications. GIAC is ANSI accredited. Certifications are available in the following key application security disciplines:

- **GWAPT** GIAC Web Application Penetration Tester
- **GWEB** GIAC Web Application Defender
- **GSSP-JAVA** GIAC Secure Software Programmer in Java
- **GSSP-.NET** GIAC Secure Software Programmer in .NET

“GIAC is the only certification that proves you have hands-on technical skills.”

- Christina Ford, Department of Commerce

“GIAC Certification demonstrates an applied knowledge versus studying a book.”

- Alan C, USMC
What’s Your Next Career Move?

SANS Technology Institute, an independent subsidiary of SANS, is now accredited by The Middle States Commission on Higher Education!

3624 Market Street | Philadelphia, PA 19104 | 267.285.5000
an institutional accrediting agency recognized by the U.S. Secretary of Education and the Council for Higher Education Accreditation.

Two unique, respected master’s degree programs:

Master of Science in Information Security Engineering

Master of Science in Information Security Management

SANS Technology Institute offers key qualities students seek in a cyber master’s program:

• World-class, cutting-edge technical courses that establish and specialize your skills
• Teaching faculty with an unparalleled reputation for industry leadership, and who bring the material to life
• Simulation and group projects that teach students to write, present and persuade effectively
• Flexibility to attend courses when and where you need them, either live in classrooms or online from home
• A reputation that helps accelerate career growth – employers will know and respect where you earned your degree

“IT’s great to learn from an organization at the forefront of both academics, and in the field.”

-Joseph Faust, MSISE Program
Critical software systems designed to enable business transactions are often at the root of many headlines about data breaches and corporate hacks. The most common attacks are often caused by simple mistakes that occur while software is being developed and deployed.

By educating everyone involved in the software development process, including developers, architects, managers, testers, business owners and partners, you reduce the chances that your organization will become a victim of today’s data security threats and ensure your team can properly build defensible applications from the start. STH.Developer provides the pinpoint software security awareness training that your team needs when they need it most, from the comfort of their own desks.

Video Demo Available
securingthehuman.org/developer
**SANS TRAINING FORMATS**

**LIVE CLASSROOM TRAINING**

**Multi-Course Training Events**
*Live Instruction from SANS’ Top Faculty, Vendor Showcase, Bonus Evening Sessions, and Networking with Your Peers*
sans.org/security-training/by-location/all

**Community SANS**
*Live Training in Your Local Region with Smaller Class Sizes*
sans.org/community

**Private Training**
*Live Training at Your Office Location*
sans.org/private-training

**Mentor**
*Live Multi-Week Training with a Mentor*
sans.org/mentor

**Summit**
*Live IT Security Summits and Training*
sans.org/summit

**ONLINE TRAINING**

**OnDemand**
*E-learning Available Anytime, Anywhere, At Your Own Pace*
sans.org/ondemand

**vLive**
*Online Evening Courses with SANS’ Top Instructors*
sans.org/vlive

**Simulcast**
*Attend a SANS Training Event Without Leaving Home*
sans.org/simulcast

**OnDemand Bundles**
*Extend Your Training with an OnDemand Bundle Including Four Months of E-learning*
sans.org/ondemand/bundles