Ingraining security into the mind of every developer.

Over 70% of today’s vulnerabilities are found in the application layer. Learn how to avoid security issues up front instead of fixing problems post design, development or deployment. Train and certify your teams and contractors in the core competencies of secure application development.

http://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 everyday 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 attack, SANS has created an Application Security Curriculum with classes in secure coding, web application security, secure code review, security testing, and software security awareness. These courses are written and taught by world-class instructors who are also everyday practitioners working on building secure applications; people who are defending their applications 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. I hope to see you online or at an upcoming SANS event!

Sincerely,

Frank Kim,
Software Security Curriculum Lead

SANS Software Security Training

SANS Software Security curriculum features courses for developers as well as security professionals who want to master the practical steps necessary for defending 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.
## Additional Software Security Courses

http://software-security.sans.org

### Secure Coding

<table>
<thead>
<tr>
<th>Language</th>
<th>Course Code</th>
<th>Course Title</th>
<th>Technology Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA</td>
<td>DEV541</td>
<td>Secure Coding in Java/JEE</td>
<td>GSSP-JAVA</td>
</tr>
<tr>
<td>.NET</td>
<td>DEV544</td>
<td>Secure Coding in .NET</td>
<td>GSSP-.NET</td>
</tr>
<tr>
<td>C &amp; C++</td>
<td>DEV543</td>
<td>Secure Coding in C &amp; C++</td>
<td></td>
</tr>
<tr>
<td>PHP</td>
<td>DEV545</td>
<td>Secure Coding in PHP</td>
<td></td>
</tr>
</tbody>
</table>

### New! Language Agnostic

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Technology Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV536</td>
<td>Secure Coding: Developing Defensible Applications</td>
<td></td>
</tr>
<tr>
<td>SEC642</td>
<td>Advanced Web App Pen Testing and Ethical Hacking</td>
<td>GWAPT</td>
</tr>
</tbody>
</table>

### Job / Role in Organization

#### Developer

- **DEV522** Defending Web Applications Security Essentials
- **DEV541** Secure Coding in Java/JEE
- **DEV544** Secure Coding in .NET
- **SEC542** Web App Pen Testing and Ethical Hacking
- **SEC642** Advanced Web App Pen Testing and Ethical Hacking

#### Senior Developer/Technical Team Lead

- **DEV522** Defending Web Applications Security Essentials
- **DEV541** Secure Coding in Java/JEE
- **DEV544** Secure Coding in .NET
- **SEC542** Web App Pen Testing and Ethical Hacking
- **SEC642** Advanced Web App Pen Testing and Ethical Hacking

#### Architect

- **DEV522** Defending Web Applications Security Essentials
- **DEV541** Secure Coding in Java/JEE
- **DEV544** Secure Coding in .NET
- **SEC542** Web App Pen Testing and Ethical Hacking
- **SEC642** Advanced Web App Pen Testing and Ethical Hacking

#### Security

- **DEV522** Defending Web Applications Security Essentials
- **SEC542** Web App Pen Testing and Ethical Hacking
- **SEC504** Hacker Techniques, Exploits and Incident Handling

#### QA/Tester

- **DEV522** Defending Web Applications Security Essentials
- **SEC542** Web App Pen Testing and Ethical Hacking

#### Manager

- **DEV522** Defending Web Applications Security Essentials

---

*All DEV courses cover the OWASP Top 10 and CWE/SANS Top 25 Software Errors*
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.

This class goes beyond classic web applications and includes coverage of Web 2.0 technologies, like AJAX and web services. We also arm you with knowledge to defend yourself against cutting-edge attackers, such as various protective HTTP headers and new generation of browser-based web application protections.

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.

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 forgery
- 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
- SQL injection and blind SQL injection attack and defense
- Cross-site scripting and defense
- HTTP response splitting defense
- Honeypot
- Intrusion detection within applications
- Incident handling of web applications
- Safe single sign-on with third party
- SOAP
- XML schema attacks
- WSDL enumeration
- XPath injection
- SAML
- XML encryption
- JSON object and security
- AJAX attack scenarios

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

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.

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 forgery
- 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
- SQL injection and blind SQL injection attack and defense
- Cross-site scripting and defense
- HTTP response splitting defense
- Honeypot
- Intrusion detection within applications
- Incident handling of web applications
- Safe single sign-on with third party
- SOAP
- XML schema attacks
- WSDL enumeration
- XPath injection
- SAML
- XML encryption
- JSON object and security
- AJAX attack scenarios

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.

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 forgery
- 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
- SQL injection and blind SQL injection attack and defense
- Cross-site scripting and defense
- HTTP response splitting defense
- Honeypot
- Intrusion detection within applications
- Incident handling of web applications
- Safe single sign-on with third party
- SOAP
- XML schema attacks
- WSDL enumeration
- XPath injection
- SAML
- XML encryption
- JSON object and security
- AJAX attack scenarios

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.
**Secure Coding in Java/JEE: Developing Defensible Applications**

**Four-Day Program • 9:00am - 5:00pm**

24 CPE/CMU Credits • Laptop Required

---

**The course to take to learn Secure Coding in Java!**

Great programmers 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. Major financial institutions and government agencies have informed their internal development teams and outsourcers that programmers must demonstrate mastery of secure coding skills and knowledge through reliable third-party testing or lose their right to work on assignments for those organizations. More software buyers are joining the movement every week. Such buyer and management demands create an immediate response from programmers, "Where can I learn what is meant by secure coding?" This unique SANS course allows you to bone up on the skills and knowledge required to prevent your applications from getting hacked.

**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 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 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.

**What You Will Learn**

- Web Application Attacks and Defenses
  - Cross-site scripting (XSS)
  - Cross-site request forgery (CSRF) - SQL injection
  - HTTP response splitting
  - Parameter manipulation
  - Directory Traversal
  - Web application proxies
  - Using Paros
- Authentication
  - Basic & forms based authentication
  - Client certificate authentication
  - Spring security
- Session Management
  - SAttacks, defense, and best practices
- Encryption
  - JSSE
  - JCA
  - Secure password storage
  - Logging & error handling
  - Race conditions
  - Java Security Manager
  - Jar Signing
  - Security in SDLC
  - Static Analysis Overview
  - Dynamic Testing Overview
  - Code Review techniques

---

**Who Should Attend:**

- Developers who want to build more secure applications
- 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

**GIAC Certification**

Package Included

- Two Practice Tests
- One Certification Exam

Offer expires 12/31/2012

**GIAC Certification**

www.giac.org

---

Software Security Curriculum | www.sans.org | 301-654-SANS (7267) | info@sans.org
The course for Secure Coding in .NET!

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 onus is still on application developers to understand the limitations of the framework and ensure that their own code is secure.

During this four-day course we will analyze the defensive strategies and technical underpinnings of the ASP.NET framework and learn where, as a developer, you can leverage defensive technologies in the framework, where you need to build security in by hand. We’ll also examine strategies for building applications that will be secure both today and in the future.

Rather than focusing on traditional web attacks from the attacker’s perspective, this class will show developers first how to think like an attacker, and will then focus on the latest defensive techniques specific to the ASP.NET environment. The emphasis of the class is a hands-on examination of the practical aspects of securing .NET applications during development.

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

What You Will Learn

- Web application attacks and defenses
  - SQL injection
    (including LDAP, XML, and Xpath injection)
  - Cross-site scripting
  - Cross-site request forgery
  - Parameter manipulation
  - Numeric overflow
- Web application proxies
  - Using Fiddler
- Assemblies
  - Strong name vs weak named
- Global Assembly Cache (GAC)
- Authentication
  - IIS / ASP.Net pluggable authentication architecture
  - Basic, digest, forms, and windows authentication
  - Authorization, OS security, and impersonation
- .NET encryption services
  - Encryption principals
  - Securing communications
  - Protecting data at rest
- .NET built in security features
  - Limited protections against cross-site request forgery, cross-site scripting, response splitting
- Exception handling and audit logging
- Security in SDL
  - Static analysis overview
  - Dynamic analysis overview
- Threat modeling
- Code-review techniques

Who Should Attend:

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

- Software developers and architects
- Senior software QA specialists
- System and security administrators
- Penetration Testers

GIAC Certification

Package Included

- Two Practice Tests
- One Certification Exam

Offer expires 12/31/2012

GIAC Certification

www.giac.org

Please check the online course description for any updates, prerequisites, laptop requirements, or special notes.
Tried and true defensive techniques for your C and C++ apps!

The C and C++ programming languages are the bedrock for most operating systems, system libraries, major network services, embedded systems and system utilities. Even though C derived languages are well understood, their flexibility coupled with inconsistencies in the standard C libraries have led to an enormous number of discovered vulnerabilities over the years. Additionally, analyzing the patching strategies when vulnerabilities are discovered often reveals that even experienced developers don’t clearly understand how these flaws occur. The unfortunate truth is that there are probably more undiscovered vulnerabilities than there are known vulnerabilities in code that we rely upon every day!

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. Students are also given a number of coding strategies that will assist them in creating more reliable code in future projects.

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

The learning doesn’t end when class is over. SANS Software Security Institute website features the AppSec 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. Visit http://software-security.sans.org – new content is added regularly, so please visit often. And don’t forget to share this information with your fellow developers and security professionals.
Hands on coding coupled with language agnostic PCI compliance training!

Secure handling of sensitive data within applications, particularly web applications, is one of the primary hot spots for security in our enterprises. Not only are there frameworks outlining the top vulnerabilities within code, but there is an emerging trend to require both secure data handling within code and secure code training for our developers. As an example, the audit procedure documents for the PCI DSS and PA DSS tell the auditor 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 and where can I get it?” This course packs a thorough explanation and examination of the top web application programming issues, which are the foundation of the PCI requirement and the OWASP Top Ten, into a two day course. What makes this course truly unique, however, is that successful attendees will be guided through the hands on development of an application that successfully defends itself against all of these flaws!

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 more than ten 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, Java, Python, or whichever language each individual student wishes to use. This will allow each student to try his 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.

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, Python, C, C++, Java, or Ruby. Each student should bring a laptop with his preferred web development environment pre-installed.

What You Will Learn

- How to bake security into the design
- The impact of the top application flaws found today
- Effective use of cryptography
- Input handling strategies
- How to design and implement a session management library
- CSRF prevention strategies
- XSS prevention strategies
- Immunity from SQL injection flaws
- Complete mediation implementations for authentication and authorization
Risk management is a top priority. The security of these assets depends on the skills and knowledge of your security team. Don’t take chances with a one-size fits all security certification. Get GIAC certified!

GIAC offers over 20 specialized certifications in security, forensics, penetration testing, web application security, audit and management.

GIAC Certifications for Application Security Professionals:

<table>
<thead>
<tr>
<th>CERT</th>
<th>CERT DESCRIPTION</th>
<th>SANS COURSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GWEB</td>
<td>Defending Web Apps</td>
<td>DEV522</td>
</tr>
<tr>
<td>GSSP-Java</td>
<td>Secure Coding in Java</td>
<td>DEV541</td>
</tr>
<tr>
<td>GSSP-.NET</td>
<td>Secure Coding in .NET</td>
<td>DEV544</td>
</tr>
<tr>
<td>GWAPT</td>
<td>Web App Pen Testing</td>
<td>SEC542</td>
</tr>
</tbody>
</table>

“GIAC is the only certification that proves you have hands-on technical skills.”
-Christina Ford, Department of Commerce

Learn more about GIAC and how to Get Certified at www.giac.org
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 Web sites 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 back-end 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.

On day one, we will study the attacker’s view of the Web as well as learn an attack methodology and how the pen-tester uses JavaScript within the test. On day two, we will study the art of reconnaissance, specifically targeted to Web applications. We will also examine the mapping phase as we interact with a real application to determine its internal structure. During day three we will continue our test by starting the discovery phase using the information we gathered on day two. We will focus on application/server-side discovery. On day four we will continue discovery, focusing on client-side portions of the application, such as Flash objects and Java applets. On day five, we will move into the final stage of exploitation. Students will use advanced exploitation methods to gain further access within the application. Day six will be a Capture the Flag event where the students will be able to use the methodology and techniques explored during class to find and exploit the vulnerabilities within an intranet site.

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 Web site designers, architects, and developers, will benefit from learning the practical art of Web application penetration testing in this class.

From the Author

Testing the security of Web applications is not as simple as just knowing what SQL injection and cross-site scripting mean. Successful testers understand that methodical, thorough testing is the best means of finding the vulnerabilities within the applications. This requires a deep understanding of how Web applications work and what attack vectors are available. This course provides that understanding by examining the various parts of a web application penetration. When teaching the class, I especially enjoy the use of real-world exercises and the in-depth exploration of Web penetration testing. – Kevin Johnson
This course is designed to teach you the advanced skills and techniques required to test web applications today. 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, which 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 due to their prevalence within modern organizations.

This information-packed advanced pen testing course will wrap up with a full-day Capture the Flag (CtF) event. This CtF event will target an imaginary organization’s web applications and will 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.

The SANS promise is that you will be able to use these ideas immediately upon returning to the office in order to better perform penetration tests of your web applications and related infrastructure. This course will enhance your exploitation and defense skill sets as well as fulfill a need to teach more advanced techniques than can be covered in the foundational course, Security 542: Web Application Penetration Testing and Ethical Hacking.

From the Author

As web applications and their mobile counterparts become more complex and hardened against attack, penetration testers need to adjust the techniques they use to evaluate the security of these systems. This includes understanding how the various targets work, their usage of encryption and web application firewalling, and how to perform vulnerability discovery and exploitation against these items. This course is designed to expand past the methodology and focus on the how when we are presented with the challenges of web penetration testing.

-Kevin Johnson
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 150 minute time limit. After taking the test, participants will receive an Assessment Report detailing the skills mastered and the areas skills could be refined. The assessment will gauge a participant's ability to avoid the most dangerous threats as 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 cyber security 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

Need help answering any of these questions?

A SANS Corporate License for training will save you money!
Secure Development Techniques

Excerpt from the WhatWorks
Top 35 Secure Development Techniques – See the full list at
www.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:

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://www.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:
   mysql_db_query("select id from users where username='$Username'")

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

2) Enable and configure Suhosin
   
   See http://www.hardened-php.net/suhosin for details about Suhosin.

3) Extract data from super globals inside validation functions only
   
   BAD:
   $UserID=$_POST['userid'];
   if ( ! is_int($userID) ) {
     $UserID=0;
   }

   BETTER:
   $UserID=get_userid('userid');
   function get_userid($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:
   print $value;

   BETTER:
   safe_out($value);

   function safe_out($value) {
     $value=htmlentities($value,ENT_QUOTES,'UTF-8');
     print $value;
   }

“Personally, I favor coding in unstructured languages like Perl and PHP for all the wrong reasons.”

-JOHANNES ULLRICH, PHD

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, 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
   public class ValidationUtility {
     public static bool TryValidateAndSanitizeLastName(string unsanitizedLastName, out string sanitizedLastName) {
       // Fail Securely
       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.

   Available AntiXSS methods: HtmlEncode(), HtmlAttributeEncode(), JavascriptEncode(), VisualBasicScriptEncode(), UrlEncode(), XmlEncode(), XmlAttributeEncode().

   ```csharp
   <div>Welcome, <%= AntiXss.HtmlEncode(Request.Form["FullName"]); %></div>
   ```

Excerpt from the WhatWorks
Top 35 Secure Development Techniques – See the full list at
www.sans.org/whatworks/poster-spring-2010.pdf
1. Assessments to gauge developers’ and contractors’ abilities to code securely in a specific language: Java and .NET
   Contact spa@sans.org for more information.

2. 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. www.sans.org/top25-programming-errors

3. 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. www.sans.org/appseccontract

4. Get GIAC Certified www.giac.org
   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
     This certification measures an individual’s understanding of Web application exploits and penetration testing methodology.
   - GWEB GIAC Web Application Defender
     Allows candidates to demonstrate mastery of the security knowledge and skills needed to deal with common web application errors that lead to most security problems.
   - GSSP-JAVA GIAC Secure Software Programmer in Java
   - GSSP-.NET GIAC Secure Software Programmer in .NET
     The GIAC Secure Software Programmer was designed for individuals who are responsible for coding secure software applications, identifying shortfalls in the security knowledge of other programmers, ensuring other programmers have adequate secure coding skills, and advanced secure programming skills.

5. Stay In Touch with the AppSec Community
   Find free resources and materials related to application security. SANS Software Security Institute Web site 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. http://software-security.sans.org

6. Application Security Street Fighter Blog
   http://software-security.sans.org/blog
Frank Kim - SANS Certified Instructor and Curriculum Lead for SANS' Software Security Curriculum

Frank Kim is the founder and principal consultant with ThinkSec and has over 14 years of experience in software development, information technology, and security. He has designed and developed applications for large healthcare, technology, insurance, and consulting companies. Frank currently focuses on security strategy and application security program development with a special interest in integrating security into the software development life cycle. Frank is the author of the SANS Institute's Secure Coding in Java course. He has spoken internationally at events like JavaOne, Devoxx, Jazoon, and UberConf and was recently named a JavaOne Rock Star.

Johannes Ullrich, PhD - SANS Senior Instructor

As chief research officer for the SANS Institute, Johannes is currently responsible for the SANS Internet Storm Center (ISC) and the GIAC Gold program. He founded DShield.org in 2000, 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. He also enjoys blogging about application security tips.

Kevin Johnson - SANS Senior Instructor

Kevin Johnson is a senior security analyst with InGuardians, LLC. Kevin came to security from a development and system administration background. He has many years of experience performing security services for Fortune 100 companies, and in his spare time he contributes to a large number of open source security projects. Kevin founded and leads the development on the Basic Analysis and Security Engine (BASE) project, the most popular web interface for the Snort intrusion detection system. Kevin is an instructor for SANS, teaching both SEC504: Hacker Techniques, Exploits, and Incident Handling and SEC542: Web App Pen Testing and Ethical Hacking. He has presented to many organizations, including Infragard, ISACA, ISSA, and the University of Florida.

Tanya Baccam - SANS Senior Instructor

Tanya is a SANS senior instructor as well as a SANS courseware author. She provides many security consulting services for clients, such as system audits, vulnerability and risk assessments, database audits, and Web application audits. Tanya has previously worked as the director of assurance services for a security services consulting firm and the
manager of infrastructure security for a healthcare organization. She also served as a manager at Deloitte & Touche in the Security Services practice. Throughout her career she’s consulted with many clients about their security architecture, including areas such as perimeter security, network infrastructure design, system audits, Web server security, and database security. She has played an integral role in developing multiple business applications and currently holds the CPA, GCFW, GCIH, CISSP, CISM, CISA, CCNA, and Oracle DBA certifications.

**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.

**James Jardine - SANS Instructor**

James Jardine is a senior security consultant at Secure Ideas and the founder of Jardine Software. James has over 12 years of software development experience with over seven years focusing on application security. His experience includes penetration testing, secure development lifecycle creation, vulnerability management, code review, and training. James is focused on helping developers build more secure applications. He has experience working with small start-ups as well as large corporations. He has worked with mobile, web, and windows development with the Microsoft .NET framework. James is a mentor for the Air Force Association’s Cyber Patriot competition. He currently holds the GSSP-NET, CSSLP, MCAD, and MCSD certifications and is located in Jacksonville, Florida.

**Mano Paul - (ISC)² Certified Instructor**

Mano Paul is an (ISC)² appointed software assurance advisor and is a seasoned veteran in the discipline of information security, software assurance and software development, with responsibilities that include designing and developing security programs from compliance to coding, security in the software development lifecycle, and providing risk management, security strategy and security awareness and education. He is the CEO and founder of SecuRisk Solutions, which specializes in security product development and consulting, and Express Certifications, a professional certification assessment and training company.
Training Events
www.sans.org/security-training/bylocation/index_all.php

Community
Community SANS
www.sans.org/community

OnSite
Live Training at Your Location
www.sans.org/onsite

Mentor
Intimate Live Instruction
www.sans.org/mentor

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

OnDemand
All the Course Content at Your Own Pace
www.sans.org/ondemand

vLive
Virtual Live Training from Your Home or Office
www.sans.org/virtual-training/vlive

Simulcast
Attend Event Training From Your Location
www.sans.org/virtual-training/event-simulcast
www.sans.org/virtual-training/custom-simulcast

SelfStudy
Independent Study with Books and MP3s
www.sans.org/selfstudy