



CWE/SANS TOP 25 Most Dangerous Programming Errors

Experts Announce Agreement on the 25 Most Dangerous Programming Errors - And How to Fix Them Agreement Will Change How Organizations Buy Software.

Project Manager: Bob Martin, MITRE

Questions: top25@sans.org

(January 12, 2009) Today in Washington, DC, experts from more than 30 US and international cyber security organizations jointly released the consensus list of the 25 most dangerous programming errors that lead to security bugs and that enable cyber espionage and cyber crime. Shockingly, most of these errors are not well understood by programmers; their avoidance is not widely taught by computer science programs; and their presence is frequently not tested by organizations developing software for sale.

The impact of these errors is far reaching. Just two of them led to more than 1.5 million web site security breaches during 2008 - and those breaches cascaded onto the computers of people who visited those web sites, turning their computers into zombies.

People and organizations that provided substantive input to the project are listed below. They are among the most respected security experts and they come from leading organizations ranging from Symantec and Microsoft, to DHS's National Cyber Security Division and NSA's Information Assurance Division, to OWASP and the Japanese IPA, to the University of California at Davis and Purdue University. The MITRE and the SANS Institute managed the Top 25 Errors initiative, but the impetus for this project came from the National Security Agency and financial support for MITRE's project engineers came from the US Department of Homeland Security's National Cyber Security Division. The Information Assurance Division at NSA and National Cybersecurity Division at DHS have consistently been the government leaders in working to improve the security of software purchased by the government and by the critical national infrastructure.

What was remarkable about the process was how quickly all the experts came to agreement, despite some heated discussion. "There appears to be broad agreement on the programming errors," says SANS Director, Mason Brown, "Now it is time to fix them. First we need to make sure every programmer knows how to write code that is free of the Top 25 errors, and then we need to make sure every programming team has processes in place to find, fix, or avoid these problems and has the tools needed to verify their code is as free of these errors as automated tools can verify."

The Office of the Director of National Intelligence expressed its support saying, "We believe that integrity of hardware and software products is a critical element of cybersecurity. Creating more secure software is a fundamental aspect of system and network security, given that the federal government and the nation's critical infrastructure depend on commercial products for business operations. The Top 25 is an important component of an overall security initiative for our country. We applaud this effort and encourage the utility of this tool through other venues such as cyber education."

Until now, most guidance focused on the 'vulnerabilities' that result from programming errors. This is helpful. The Top 25, however, focuses on the actual programming errors, made by developers that create the vulnerabilities. As important, the Top 25 web site provides detailed and authoritative information on mitigation. "Now, with the Top 25, we can spend less time working with police after the house has been robbed and instead focus on getting locks on the doors before it happens." said Paul Kurtz, a principal author of the US National Strategy to Secure Cyberspace and executive director of the Software Assurance Forum for Excellence in Code (SAFECode).

What You Will Find In This Announcement:

- [Which People and Organizations Made Substantive Contributions to the Top 25 Errors List?](#)
- [How Will the Top 25 Errors Be Used?](#)
- [How Important Are the Top 25 Errors?](#)
- [What Errors Are Included in the Top 25?](#)

- [Resources to Help Organizations Eliminate The Errors](#)

Which People and Organizations Made Substantive Contributions to the Top 25 Errors List?

Robert C. Seacord, CERT
Pascal Meunier, CERIAS, Purdue University
Matt Bishop, University of California, Davis
Kenneth van Wyk, KRvW Associates
Masato Terada, Information-Technology Promotion Agency (IPA), (Japan)
Sean Barnum, Cigital, Inc.
Mahesh Saptarshi and Cassio Goldschmidt, Symantec Corporation
Adam Hahn, MITRE
Jeff Williams, Aspect Security
Carsten Eiram, Secunia
Josh Drake, iDefense Labs at VeriSign, Inc.
Chuck Willis, MANDIANT
Michael Howard, Microsoft
Bruce Lowenthal, Oracle Corporation
Mark J. Cox, Red Hat Inc.
Jacob West, Fortify Software
Djenana Campara, Hatha Systems
James Walden, Northern Kentucky University
Frank Kim, ThinkSec
Chris Eng and Chris Wysopal, Veracode, Inc.
Ryan Barnett, Breach Security
Antonio Fontes, New Access SA, (Switzerland)
Mark Fioravanti II, Missing Link Security Inc.
Ketan Vyas, Tata Consultancy Services (TCS)
Lindsey Cheng, Ian Peters and Tom Burgess, Secured Sciences Group, LLC
Hardik Parekh and Matthew Coles, RSA - Security Division of EMC Corporation
Mouse
Ivan Ristic
Apple Product Security
Software Assurance Forum for Excellence in Code (SAFECode)
Core Security Technologies Inc.
Depository Trust & Clearing Corporation (DTCC)
The working group at the first OWASP ESAPI Summit
National Security Agency (NSA) Information Assurance Division
Department of Homeland Security (DHS) National Cyber Security Division

Robert Martin, CWE Project Leader at MITRE heralded the effort of these contributors by saying, "It is gratifying to see the amount of collaboration and energy that all these serious, security-savvy people invested in making this list as accurate and authoritative as it can be. Very impressive!"

How Will the Top 25 Errors Be Used?

The Top 25 Errors will have four major impacts:

- Software buyers will be able to buy much safer software.
- 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.
- Employers will be able to ensure they have programmers who can write more secure code.

First, software buyers will be able to buy much safer software.

Buyers will require that software vendors certify in writing that the code they are delivering is free of these 25 programming errors. Certification shifts responsibility to the vendor for correcting the errors and for any damage caused by those errors. The standard procurement language under development by the State of New York and other state governments already is being

adjusted to use the Top 25 Errors. Over time the multi-national Common Criteria program may also adopt the Top 25 as one approach for ensuring code purchased by the US government is free of the Top 25 errors.

Second, programmers will have tools that consistently measure the security of the software they are writing.

Software testing tools will use the Top 25 in their evaluations and provide scores for the level of secure coding in software being tested. In parallel with this announcement, on January 12, one of the leading software testing vendors is announcing that its software will be able to test for and report on the presence of a large fraction of the Top 25 Errors. Application development teams will use such testing software during the development process.

Colleges will be able to teach secure coding more confidently.

Colleges and others who prepare programmers will use the Top 25 Errors as a foundation for curriculum that ensures their students know how to avoid the critical programming errors. One of the colleges that participated in developing the Top 25, UC Davis, has already established a secure coding clinic where student-written software is reviewed for the key programming errors that lead to critical security vulnerabilities. The Top 25 enables the clinic to prioritize errors in its review. Other colleges are beginning to emulate the secure coding clinics.

Employers will be able to ensure they have programmers who can write more secure code.

Employers will use the Top 25 Errors list as a guide for evaluating and improving skills of programmers they hire and of outsourced programming talent. More than 100 large employers are already using a common assessment tool called the GSSP (GIAC Secure Software Programmer) to measure secure coding skills. The GSSP exams are being reviewed in an effort to fully incorporate and highlight mastery of programming knowledge needed to find and eliminate or avoid the Top 25. More data on the GSSP may be found at <http://www.sans-ssi.org/> and organizations with at least 500 programmers may have up to 100 of those programmers' secure coding skills assessed confidentially and at no cost. Email spa@sans.org to get that started.

Courses are available that teach secure coding skills to programmers in C/C++, in Java, and in .NET languages. Information at <http://www.sans-ssi.org/courses/>

How Important Are the Top 25 Errors?

We asked several of the participants why they thought this effort was important enough to merit a significant amount of their time and expertise. Here are a few of their answers. More are at the end of the announcement.

National Security Agency's Information Assurance Directorate

"The publication of a list of programming errors that enable cyber espionage and cyber crime is an important first step in managing the vulnerability of our networks and technology. There needs to be a move away from reacting to thousands of individual vulnerabilities, and to focus instead on a relatively small number of software flaws that allow vulnerabilities to occur, each with a general root cause. Such a list allows the targeting of improvements in software development practices, tools, and requirements to manage these problems earlier in the life cycle, where they can be solved on a large scale and cost-effectively."

-Tony Sager, National Security Agency's Information Assurance Directorate

US Department of Energy:

"The CWE/SANS Top 25 effort is extremely valuable and will provide many organizations with a tangible way to begin addressing software security problems."

- Michael Klosterman, SCADA Operations, Western Area Power Association, US Department of Energy

Depository Trust:

"The CWE-SANS Top 25 Errors is a vital tool for organizations that believe in a risk-based approach to software security enabling them to assess the specific vulnerabilities identified in their environments compared with a composite perspective of risk from industry recognized experts."

- Jim Routh, CISO, The Depository Trust & Clearing Corporation

Microsoft:

"The 2009 CWE/SANS Top 25 Programming Errors project is a great resource to help software developers identify which security vulnerabilities are the most important to understand, prevent and fix."

- Michael Howard, Principal Security Program Manager, Security Development Lifecycle Team, Microsoft Corp.

OWASP Foundation:

"When facing a huge application portfolio that could contain many thousands of instances of over 700 different types of

weaknesses, knowing where to start is a daunting task. Done right, stamping out the CWE Top 25 can not only make you significantly more secure but can cut your software development costs."

- Jeff Williams, Aspect Security CEO and The OWASP Foundation Chair

Symantec:

"The 2009 CWE/SANS Top 25 Programming Errors reflects the kinds of issues we've seen in application software and helps provide us with actionable direction to continuously improve the security of our software."

- Wesley H. Higaki, Director, Software Assurance, Office of the CTO, Symantec Corporation

Software Assurance Consortium:

"As an advocate for the consumer, this is viewed as a giant step forward in providing security for all users. It increases awareness of the various levels of secure software by highlighting its effects on our daily use of all software products. The CWE/SANS Top 25 effort adds the capability to our tool box which in turn aids the SwAC in our mission to bring together Industry and Government to transform the security and dependability of all software products."

- Dan Wolf, Director, Software Assurance Consortium.

EMC:

"The Top 25 List puts a powerful tool into the hands of the programmers along with every person involved in designing and developing software. The simple fact that such a list now exists will allow software assurance to be practiced more effectively."

- Dan Reddy, Consulting Product Manager, EMC Product Security Office

Purdue:

"The CWE Top 25 should be watched because targeting the most troublesome programming mistakes can potentially reduce the occurrence of vulnerabilities and our exposure at a national level, while diminishing our undesirable dependence on patches."

- Pascal Meunier, CERIAS, Purdue University

Secunia:

"This Top 25 is without a doubt one of the most useful compilations of common coding mistakes leading to vulnerabilities in software. The list, which has been created based on feedback from many experts in the security industry, focuses on selection criteria like severity and prevalence, thus covering a broad range of the most critical errors commonly introduced in applications today. The Top 25 is compiled in a easy-to-read and entertaining language and does not only provide a good understanding of common coding mistakes, but also how to avoid them. I can therefore highly recommend this read to anyone involved in software design to ensure that they won't make the same mistakes in 2009 as they've made previously."

- Carsten Eiram, Chief Security Specialist, Secunia.

Ken van Wyk:

"This list of programming errors should be enormously useful to the community. It serves to help us all get our collective "arms around" understanding the most common security defects in our code, just as the OWASP Top 10 helps us understand the attacks against those defects."

- Kenneth R. van Wyk, KRvW Associates, LLC

Veracode:

"A prioritized list of security issues is the starting point to make software security practical in the business world of resource constraints and ship dates. The Top 25 list gives developers a minimum set of coding errors that must be eradicated before software is used by customers."

- Chris Wysopal, Co-Founder and CTO of Veracode, Inc.

Core Security Technologies:

"This is the first serious attempt at building a taxonomy of software security weaknesses and flaws with an emphasis on the practical application of identifying, preventing and fixing or mitigating the issues they pose. It is a necessary and long overdue step towards creating a common language for the software development and security communities in need of a more rational way to address what are currently the most urgent and relevant software security problems."

- Ivan Arce, CTO of Core Security Technologies Inc.

Breach Security:

"The CWE/SANS Top 25 List is an excellent tactical resource for organizations to prioritize and remediate the root causes of today's successful attacks. This should be required reading for all developers as it is a "Cliff Notes" version of essential secure coding principles."

- Ryan C. Barnett, Director of Application Security Research, Breach Security

McAfee:

"The 2009 CWE/SANS Top 25 Programming Errors effort is right on target. By educating software developers on the most important issues and showing them how to avoid writing security bugs, this effort will help programmers correct code issues before they become security problems."

- Kent Landfield, Director, Risk and Compliance Security Research, McAfee, Inc.

Ounce Lab:

"Let's use this list as a way to jumpstart the solutions - make 2009 a year to make things happen and solve these problems that have been around way too long. Far too many solutions exist out there to help address these all-too-common errors. Start using this list to secure your software today because if the last few years have been any indication, tomorrow is already too late."

- Ryan Berg, Co-Founder and Chief Scientist, Ounce Labs

Grammatech:

"Bugs in software are a plague on our profession and bad for business. They are inevitable, yet understanding of which bugs are most important is often gained the hard and expensive way when they show up in the field. The CWE/SANS Top 25 effort will raise awareness of the huge variety of different kinds of defects that can occur, and will help programmers focus on those that matter most to application quality and security."

- Paul Anderson - Vice President of Engineering, Grammatech Inc.

What Errors Are Included in the Top 25?

The Top 25 Errors are listed below in three categories:

- [Category: Insecure Interaction Between Components \(9 errors\)](#)
- [Category: Risky Resource Management \(9 errors\)](#)
- [Category: Porous Defenses \(7 errors\)](#)

Clicking "MORE" in any of the listings takes you to the relevant spot in the MITRE CWE site where you will find the following:

- links to the full CWE entry data,
- data fields for weakness prevalence and consequences,
- remediation cost,
- ease of detection,
- attack frequency and attacker awareness
- related CWE entries
- related patterns of attack for this weakness.

Each entry at the Top 25 Errors site also includes fairly extensive prevention and remediation steps that developers can take to mitigate or eliminate the weakness.

CATEGORY: Insecure Interaction Between Components

CWE-20: Improper Input Validation

It's the number one killer of healthy software, so you're just asking for trouble if you don't ensure that your input conforms to expectations... For more see: <http://cwe.mitre.org/top25/#CWE-20>

CWE-116: Improper Encoding or Escaping of Output

Computers have a strange habit of doing what you say, not what you mean. Insufficient output encoding is the often-ignored sibling to poor input validation, but it is at the root of most injection-based attacks, which are all the rage these days... For more see: <http://cwe.mitre.org/top25/#CWE-116>

CWE-89: Failure to Preserve SQL Query Structure (aka 'SQL Injection')

If attackers can influence the SQL that you use to communicate with your database, then they can... For more see: <http://cwe.mitre.org/top25/#CWE-89>

CWE-79: Failure to Preserve Web Page Structure (aka 'Cross-site Scripting')

Cross-site scripting (XSS) is one of the most prevalent, obstinate, and dangerous vulnerabilities in web applications...If you're not careful, attackers can... For more see: <http://cwe.mitre.org/top25/#CWE-79>

CWE-78: Failure to Preserve OS Command Structure (aka 'OS Command Injection')

When you invoke another program on the operating system, but you allow untrusted inputs to be fed into the command string that you generate for executing the program, then you are inviting attackers... For more see: <http://cwe.mitre.org/top25/#CWE-78>

CWE-319: Cleartext Transmission of Sensitive Information

If your software sends sensitive information across a network, such as private data or authentication credentials, that information crosses many... For more see: <http://cwe.mitre.org/top25/#CWE-319>

CWE-352: Cross-Site Request Forgery (CSRF)

With cross-site request forgery, the attacker gets the victim to activate a request that goes to your site. Thanks to scripting and the way the web works in general, the victim... For more see: <http://cwe.mitre.org/top25/#CWE-352>

CWE-362: Race Condition

Attackers will consciously look to exploit race conditions to cause chaos or get your application to cough up something valuable... For more see: <http://cwe.mitre.org/top25/#CWE-362>

CWE-209: Error Message Information Leak

If you use chatty error messages, then they could disclose secrets to any attacker who dares to misuse your software. The secrets could cover a wide range of valuable data... For more see: <http://cwe.mitre.org/top25/#CWE-209>

CATEGORY: Risky Resource Management

CWE-119: Failure to Constrain Operations within the Bounds of a Memory Buffer

Buffer overflows are Mother Nature's little reminder of that law of physics that says if you try to put more stuff into a container than it can hold, you're... For more see: <http://cwe.mitre.org/top25/#CWE-119>

CWE-642: External Control of Critical State Data

There are many ways to store user state data without the overhead of a database. Unfortunately, if you store that data in a place where an attacker can... For more see: <http://cwe.mitre.org/top25/#CWE-642>

CWE-73: External Control of File Name or Path

When you use an outsider's input while constructing a filename, you're taking a chance. If you're not careful, an attacker could... [href="http://cwe.mitre.org/top25/#CWE-73"](http://cwe.mitre.org/top25/#CWE-73)

CWE-426: Untrusted Search Path

If a resource search path is under attacker control, then the attacker can modify it to point to resources of the attacker's choosing. This causes the software to access the wrong resources at the wrong time... For more see: <http://cwe.mitre.org/top25/#CWE-426>

CWE-94: Failure to Control Generation of Code (aka 'Code Injection')

For ease of development, sometimes you can't beat using a couple lines of code to employ lots of functionality. It's even cooler when... For more see: <http://cwe.mitre.org/top25/#CWE-94>

CWE-494: Download of Code Without Integrity Check

You don't need to be a guru to realize that if you download code and execute it, you're trusting that the source of that code isn't malicious. But attackers can perform all sorts of tricks... For more see: <http://cwe.mitre.org/top25/#CWE-494>

CWE-404: Improper Resource Shutdown or Release

When your precious system resources have reached their end-of-life, you need to... For more see: <http://cwe.mitre.org/top25/#CWE-404>

CWE-665: Improper Initialization

Just as you should start your day with a healthy breakfast, proper initialization helps to ensure... For more see: <http://cwe.mitre.org/top25/#CWE-665>

CWE-682: Incorrect Calculation

When attackers have some control over the inputs that are used in numeric calculations, this weakness can lead to vulnerabilities. It could cause you to make incorrect security decisions. It might cause you to... For more see: <http://cwe.mitre.org/top25/#CWE-682>

CATEGORY: Porous Defenses

CWE-285: Improper Access Control (Authorization)

If you don't ensure that your software's users are only doing what they're allowed to, then attackers will try to exploit your improper authorization and... For more see: <http://cwe.mitre.org/top25/#CWE-285>

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

You may be tempted to develop your own encryption scheme in the hopes of making it difficult for attackers to crack. This kind of grow-your-own cryptography is a welcome sight to attackers... For more see: <http://cwe.mitre.org/top25/#CWE-327>

CWE-259: Hard-Coded Password

Hard-coding a secret account and password into your software's authentication module is... For more see: <http://cwe.mitre.org/top25/#CWE-259>

CWE-732: Insecure Permission Assignment for Critical Resource

If you have critical programs, data stores, or configuration files with permissions that make your resources accessible to the world - well, that's just what they'll become... For more see: <http://cwe.mitre.org/top25/#CWE-732>

CWE-330: Use of Insufficiently Random Values

If you use security features that require good randomness, but you don't provide it, then you'll have attackers laughing all the way to the bank... For more see: <http://cwe.mitre.org/top25/#CWE-330>

CWE-250: Execution with Unnecessary Privileges

Spider Man, the well-known comic superhero, lives by the motto "With great power comes great responsibility." Your software may need special privileges to perform certain operations, but wielding those privileges longer than necessary can be extremely risky... For more see: <http://cwe.mitre.org/top25/#CWE-250>

CWE-602: Client-Side Enforcement of Server-Side Security

Remember that underneath that fancy GUI, it's just code. Attackers can reverse engineer your client and write their own custom clients that leave out certain inconvenient features like all those pesky security controls... For more see: <http://cwe.mitre.org/top25/#CWE-602>

Resources to Help Eliminate The Top 25 Errors

The TOP 25 Errors List will be updated regularly and will be posted at both the SANS and MITRE sites

www.sans.org/top25
cwe.mitre.org/top25/

MITRE maintains the CWE (Common Weakness Enumeration) web site, with the support of the US Department of Homeland Security's National Cyber Security Division, presenting detailed descriptions of the top 25 programming errors along with authoritative guidance for mitigating and avoiding them. That site also contains data on more than 700 additional programming errors, design errors and architecture errors that can lead to exploitable vulnerabilities. cwe.mitre.org/

SANS maintains a series of assessments of secure coding skills in three languages along with certification exams that allow programmers to determine gaps in their knowledge of secure coding and allows buyers to ensure outsourced programmers have sufficient programming skills. Organizations with more than 500 programmers can assess the secure coding skills of up to 100 programmers at no cost.

Email spa@sans.org for details

And see www.sans-ssi.org/certification/ for the GSSP Blueprints

SAFECode - The Software Assurance Forum for Excellence in Code (members include EMC, Juniper, Microsoft, Nokia, SAP and Symantec) has produced two excellent publications outlining industry best practices for software assurance and providing practical advice for implementing proven methods for secure software development.

http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf

http://www.safecode.org/publications/SAFECode_Dev_Practices1108.pdf

Nearly a dozen software companies offer automated tools that test programs for these errors. SANS maintains case studies of user experience with these and other security tools at www.sans.org/whatworks.

New York State has produced draft procurement standards to allow companies to buy software with security baked in.

Draft New York State procurement language will be posted at www.sans.org/appsecontract.

For additional information on any of these:

SANS: Mason Brown, mbrown@sans.org

MITRE: Bob Martin, ramartin@mitre.org

MITRE: Steve Christey, coley@mitre.org

SANS Software Security Institute (SSI) Courses

- [Web Application Pen Testing Hands-On Immersion : Developer 538](#)
- [Web App Penetration Testing and Ethical Hacking : Security 542](#)
- [Intro to Web Application Security : Developer 319](#)
- [Web Application Security Essentials : Developer 422](#)
- [Secure Coding in Java/JEE: Developing Defensible Applications : Developer 541](#)
- [Secure Coding for PCI Compliance : Developer 536](#)
- [Defensible .NET : Developer 616](#)
- [Exploiting Regular Expressions to Process Text : Security 651](#)
- [AJAX and Web Services Security Overview : Security 426](#)
- [Java Quality Assurance, Security Testing and Auditing : Audit 428](#)
- [Secure Web Services for Managers : Management 431](#)
- [Security Policy & Awareness : Management 524](#)
- [Software Security Awareness : Developer 304](#)

© 2000-2009 The SANS™ Institute

SANS Web Privacy Policy: www.sans.org/privacy.php - Web Contact: webmaster@sans.org

SANS Press Room: www.sans.org/press / Policy On SANS Trademark Usage