

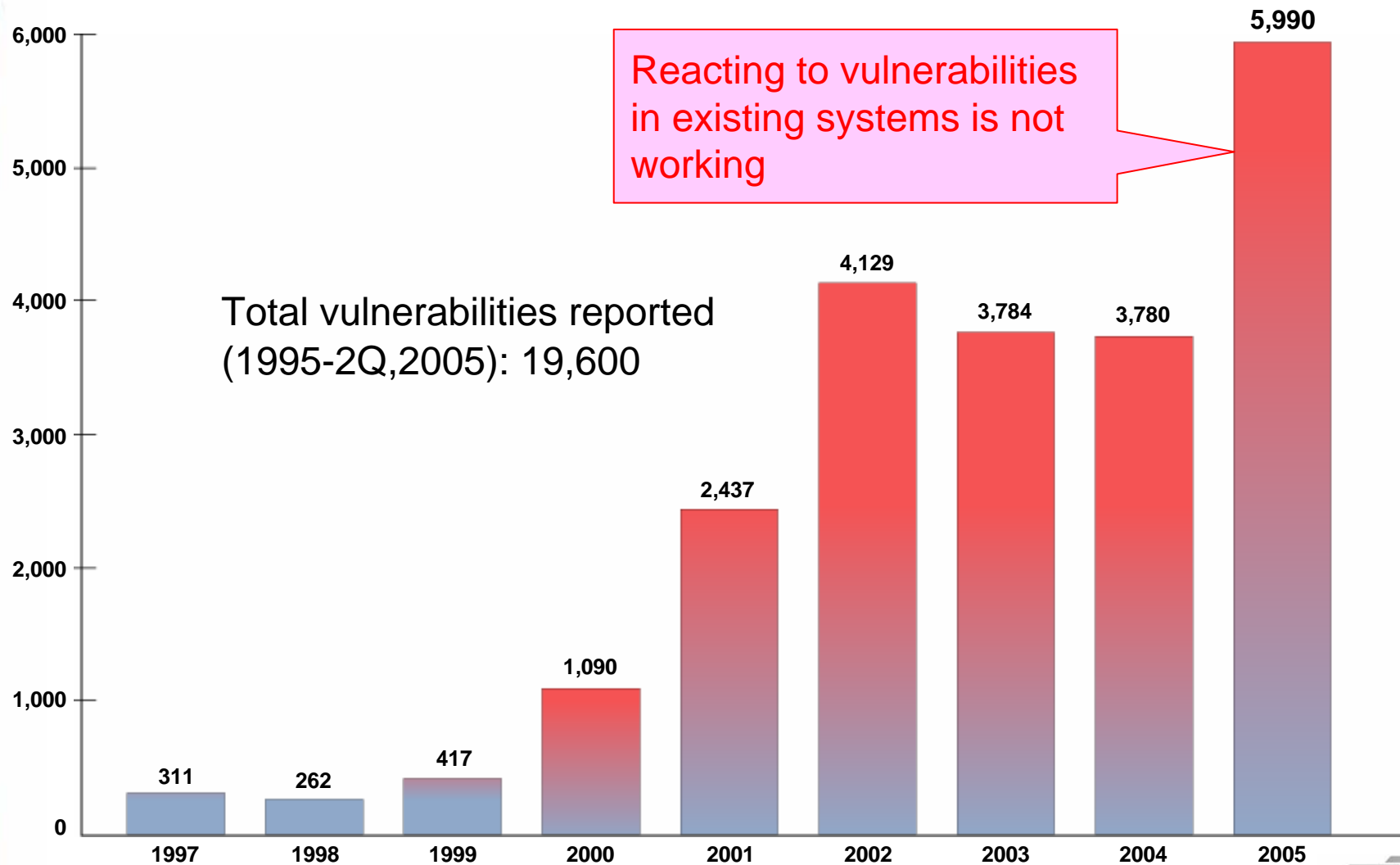


# CERT

## Secure Coding Initiative

Robert C. Seacord

# Problem Statement



# Secure Coding Initiative

---

Work with **software developers** and **software development organizations** to eliminate vulnerabilities resulting from coding errors before they are deployed.

- **Reduce** the number of vulnerabilities to a level where they can be handled by computer security incident response teams (CSIRTs)
- **Decrease** remediation costs by eliminating vulnerabilities *before* software is deployed

# Overall Thrusts

---

Advance the **state of the practice** in secure coding

Identify common programming errors that lead to software vulnerabilities

Establish standard secure coding practices

Educate software developers

# Current Capabilities

---

Secure Coding in C and C++

- Addison-Wesley book
- Training

Secure coding web pages at

[www.cert.org/secure-coding/](http://www.cert.org/secure-coding/)

COM Object tester

Managed string library

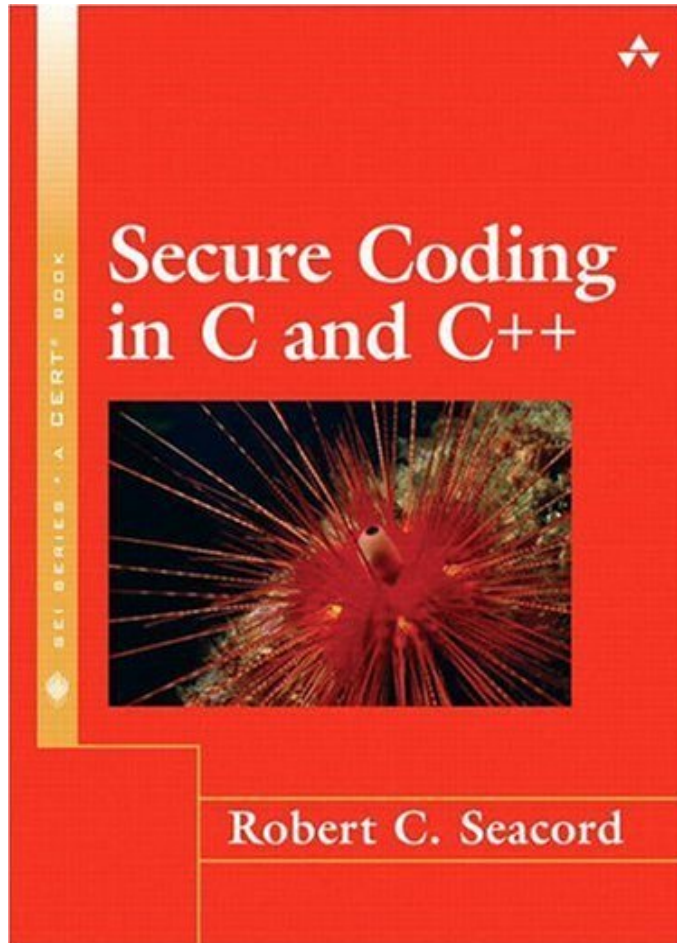
Involvement in international standards activities:

- ISO/IEC JTC1/SC22/WG14 C programming language international standardization working group
- ISO/IEC JTC1/SC22 OWG Vulnerabilities

Build Security In website

<https://buildsecurityin.us-cert.gov/>

# Secure Coding in C and C++



Practical examples of

- Insecure coding practices
- Exploitable vulnerabilities
- Effective mitigation strategies

Audience includes

- C/C++ programmers
- software project managers
- computer science students
- security analysts

Used as textbook by

- Columbia University
- Penn State
- Santa Clara University
- University of Pittsburgh

For more info see: <http://www.cert.org/books/secure-coding/>

# Secure Coding in C/C++ Training

---

Secure Coding in C and C++ provides **practical guidance** on secure programming in C and C++

- focuses on security issues intrinsic to the C and C++ programming languages and associated libraries
- provides a detailed explanation of common programming errors
- describes how errors can lead to vulnerable code
- evaluates available mitigation strategies

Useful to anyone involved in developing secure C and C++ programs regardless of the specific application

**Currently being licensed to SEI Partners**



## Options

### Advisories

[US-CERT  
Vulnerability  
Notes  
Database](#)

[Incident Notes](#)

[Current Activity](#)

## Related

[Tech Tips](#)

[AirCERT](#)

[Employment  
Opportunities](#)

## more links

[CERT Statistics](#)

[Vulnerability  
Disclosure Policy](#)

[CERT  
Knowledgebase](#)

[System  
Administrator  
courses](#)

[CSIRT courses](#)

[Other Sources of  
Security  
Information](#)

[Channels](#)

## Related Sites

## Secure Coding

The primary cause of commonly exploited software vulnerabilities is software defects that could have been avoided. Through our analysis of thousands of vulnerability reports, the CERT/CC has observed that most of them stemmed from a relatively small number of root causes. If we can identify the root causes of vulnerabilities and develop secure coding practices for illustration, software producers may be able to take practical steps to prevent introduction of vulnerabilities into deployed software systems.

Toward that goal, our systematic approach has led us to identify program errors most likely to cause security breaches. We have also identified some good practices to avoiding certain categories of vulnerabilities. Software producers can use this information as they develop strategies to avoid vulnerabilities when they code new software.

### Call for Papers

CERT is sponsoring the Secure Software Architecture, Design, Implementation and Assurance Minitrack at this year's Hawaii International Conference on System Sciences. The [call for papers](#) has been announced.

### Current Projects

- **Managed string library** - We are developing a managed string library for C that provides an alternative to standard C style strings and existing string operations. These string functions are based on a dynamic string type that automatically allocates memory as required. If you are interested in alpha-testing the library, please contact us at [secure-coding@cert.org](mailto:secure-coding@cert.org).

### Related Publications

- [Secure Coding in C and C++](#)
- [Software Vulnerabilities in Java](#)

### Presentations

- [Secure Coding in C and C++ - Tutorial](#)
- [Best Practices for Secure Coding](#)
- [Secure Coding in C and C++: A Look at Common Vulnerabilities](#)

### Related Vulnerabilities

Design and coding defects frequently cause security problems:

- [Buffer Overflows](#)
- [Format String Errors](#)
- [Integer Overflows](#)
- [Race Conditions](#)
- [Memory Management Errors](#)

### Other Resources

- [Build Security In](#)
- [ISO/IEC JTC1/SC22/WG14 - C](#)  
The programming language international standardization working group offers a wide range of information, including
  - [Specification for Safer, More Secure C Library Functions](#) - a draft technical report
  - [ISO/IEC 9899:TC2 - Programming languages - C - the current C programming language standard](#)

# COM Object Tester

---

Automatically tests all COM objects installed on a particular platform that are allowed to execute or can be loaded in Internet Explorer

Generates a detailed report describing each defect that can be a vulnerability

# COM Tester Approach

---

Programmatically identifies all installed COM objects that are allowed to execute in Internet Explorer where

- The Kill Bit is not set
- The object is marked safe for scripting or initialization
- The object has type information to get methods and properties

# COM Object Testing

---

Tests all non-restricted interfaces (methods, properties and initialization parameters) by passing them potentially unexpected data.

- Large strings (buffer overrun)
- Negative values
- Zero values
- Large negative floating point numbers
- NULL values
- Unexpected Boolean values (i.e., 10,000)

# COM Tester Reporting

---

If an exception is generated while invoking a COM interface with unexpected values the following information is reported:

- The file name, description and version information of the COM object
- A detailed trace of exercised properties and methods invoked leading up to exception
- Detailed information describing the exception

Will also generate HTML test case that can be loaded into IE to reproduce fault

# COM Object Tester Status

---

Since November 2005

- Discovered **22** vulnerabilities that allow a malicious website to run arbitrary code
- Discovered **50+** denial-of-service vulnerabilities

Source code and documentation available

License agreement

- requires non-disclosure
- prohibits further distribution

# Managed String Library

---

Developed in response to the need for a string library that can

- improve the quality and security of newly developed C-language programs
- eliminate obstacles to widespread adoption and possible standardization.

Eliminates the possibility of vulnerabilities resulting from string operations including

- Buffer overflow
- String truncation
- Null-termination errors

# Managed String Library Availability

---

Managed string library specification has been

- published as an SEI technical report
- proposed to ISO/IEC WG14 C standard committee for publication as an ISO/IEC technical report

A managed string course was offered at the Software Security Summit in February, 2006.

Source code for a "proof-of-concept" implementation of the managed string library is available for download

The reports, course, and library can be accessed at:

<http://www.cert.org/secure-coding/managedstring.html>

# Current and Projected Efforts

---

## CERT Secure Coding Standards

- C Programming Language
- C++
- Community development process

## Software Assurance Test and Analysis Tools

C/C++ Secure Programmer certification program

Training courses

# CERT Secure Coding Standards

---

Identifies coding practices that can be used to improve the security of software systems under development

Coding practices are included as either rules or recommendations

Development of Secure Coding Standards is a community effort

# Rules and Recommendations

---

Coding practices are defined to be **rules** when

- Violation of the coding practice will result in a security flaw that may result in an exploitable vulnerability.
- There is an enumerable set of exceptional conditions (or no such conditions) where violating the coding practice is necessary to ensure the correct behavior for the program.
- Conformance to the coding practice can be verified.

Coding practices are defined to be **recommendations** when

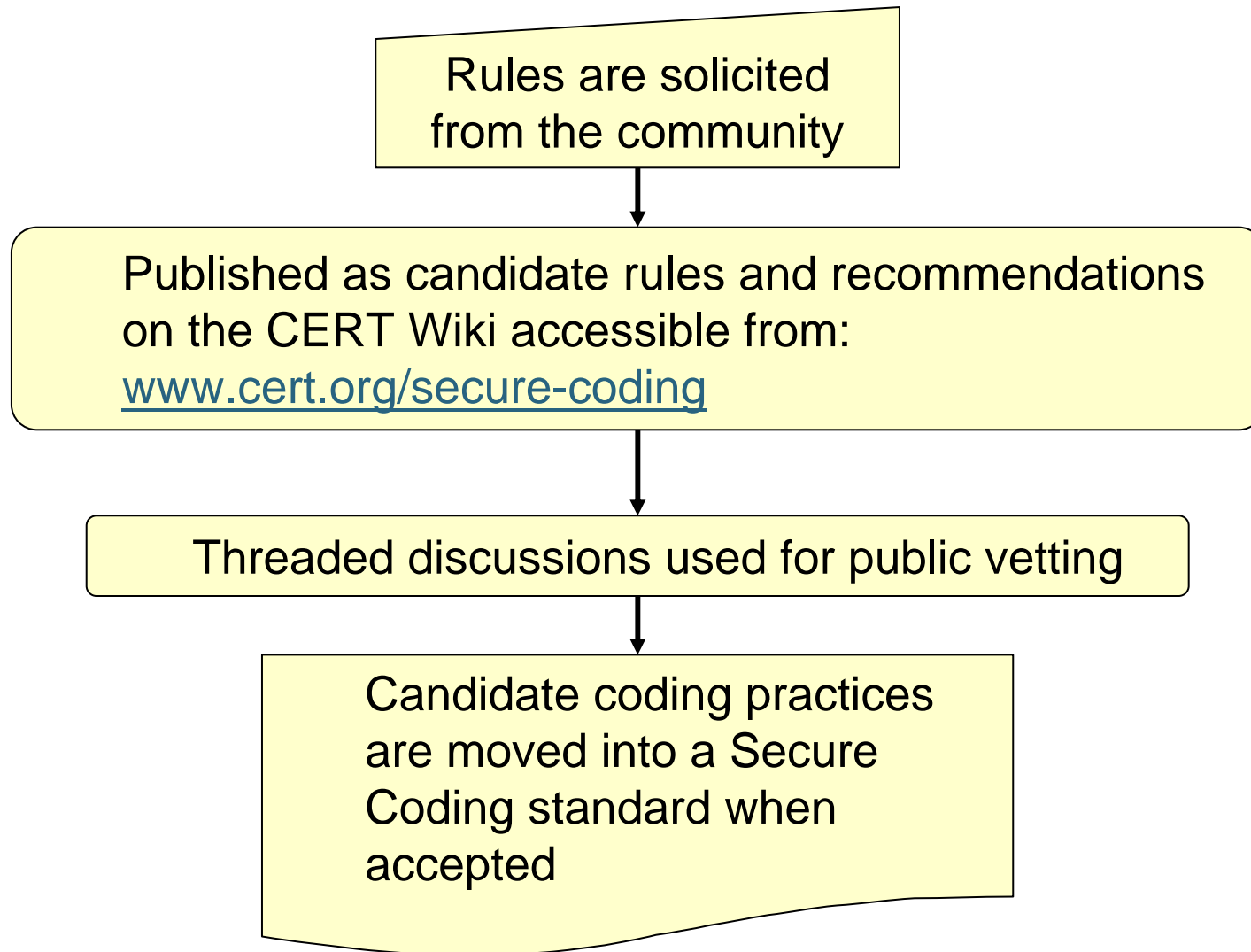
- Application of the coding practice is likely to improve system security.
- One or more of the requirements necessary for a coding practice to be considered a rule cannot be met.

Rules need to be followed to claim **compliance**.

Recommendations are **guidelines** or **suggestions**.

# Community Development Process

---



# Software Assurance Test and Analysis Tools

---

Develop automated test and analysis tools

- Reduce the time and effort (cost) required to discover and repair vulnerabilities
- Help developers identify vulnerabilities in their products during development and test

Develop tool kits that reduce the time and effort required to build custom test and analysis tools

# Secure Programmer Certification

Provides a metric for assessing an organizations ability to develop secure software systems



To be CERTified as a Secure C/C++ Programmer an applicant must

- Complete a 5-day course in Secure Coding in C and C++ from the Software Engineering Institute (SEI) or its licensees
- Submit a Certification Application package
- Pass an individual assessment examination administered by the SEI

# Future Directions

---

Provide similar products for other languages and programming environments

- C++/CLI
- C#
- Java
- Web Development
- Language independent

# Questions



# For More Information

---

## Visit the CERT® web site

<http://www.cert.org/secure-coding/>

## Contact Presenter

Robert C. Seacord      [rsc@cert.org](mailto:rsc@cert.org)

## Contact CERT Coordination Center

Software Engineering Institute  
Carnegie Mellon University  
4500 Fifth Avenue  
Pittsburgh PA 15213-3890

Hotline: **412-268-7090**

**CERT/CC personnel answer 8:00 a.m.–5:00 p.m.  
and are on call for emergencies during other hours.**

Fax:      **412-268-6989**

E-mail:    **cert@cert.org**