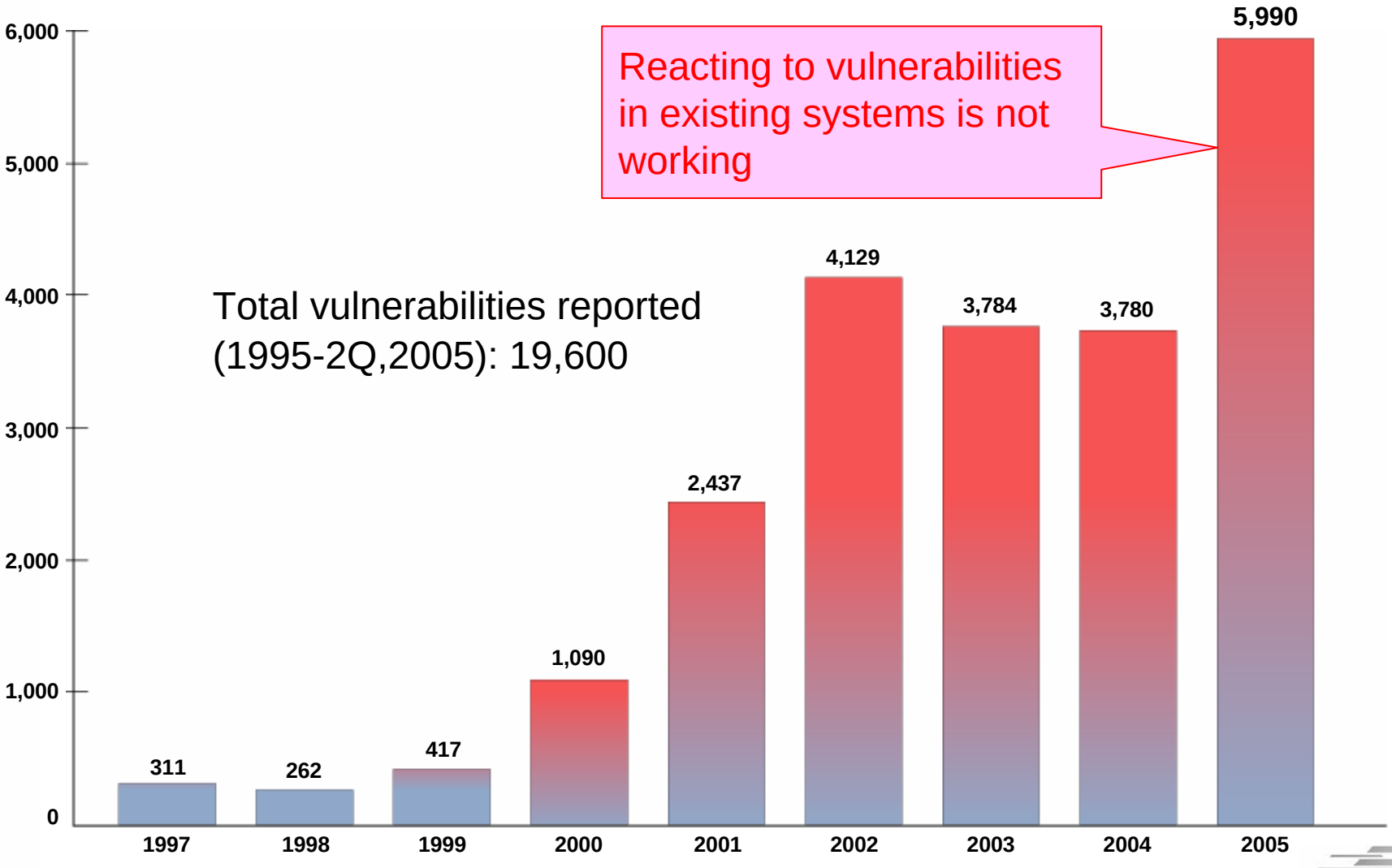
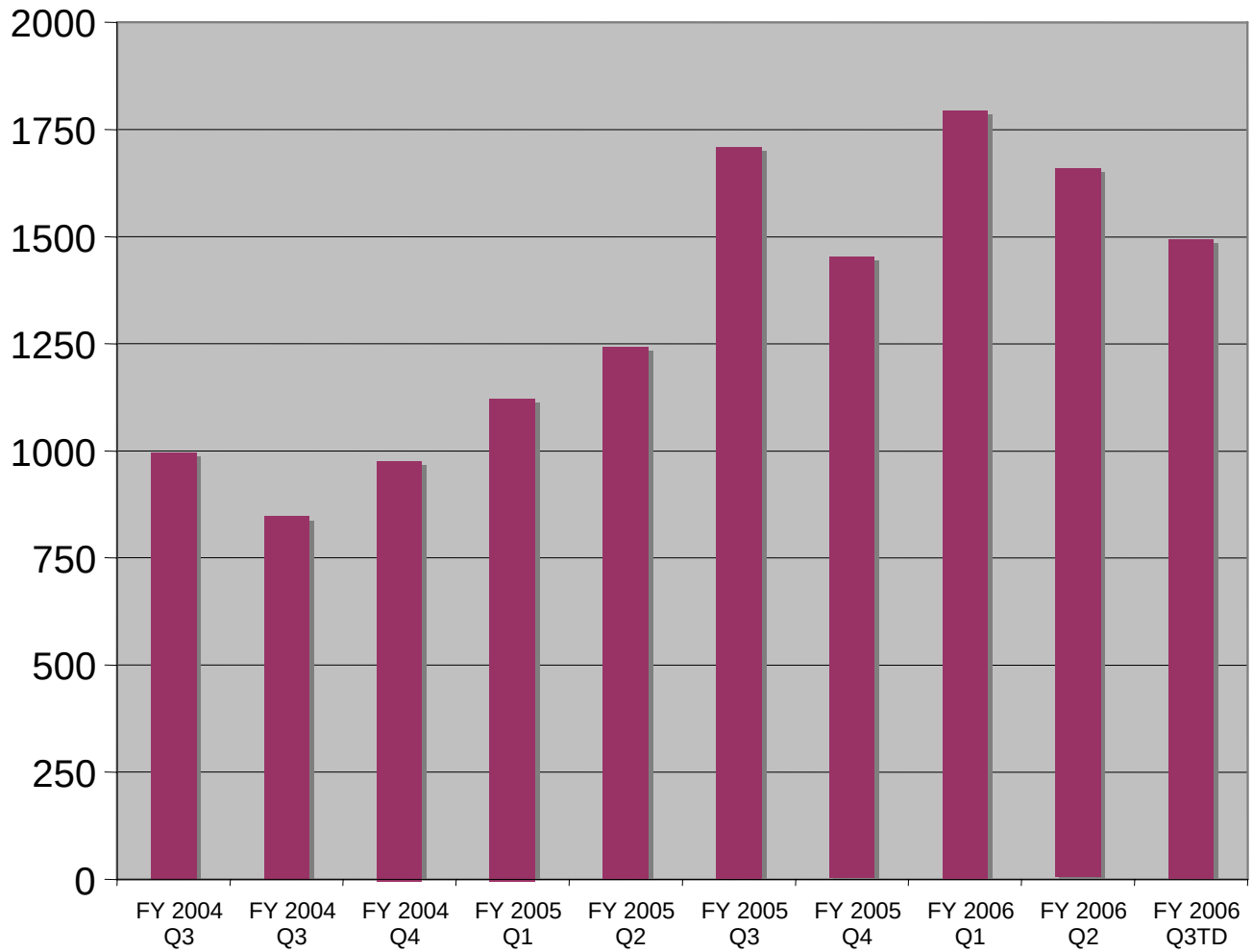
CERT Secure Coding Standards

Robert C. Seacord

Problem Statement



Recent Trends Are No Different



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

CERT Secure Coding Standards

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

Coding practices are classified as either rules or recommendations

- Rules need to be followed to claim compliance.
- Recommendations are guidelines or suggestions.

Development of Secure Coding Standards is a community effort

Rules

Coding practices are defined as rules when

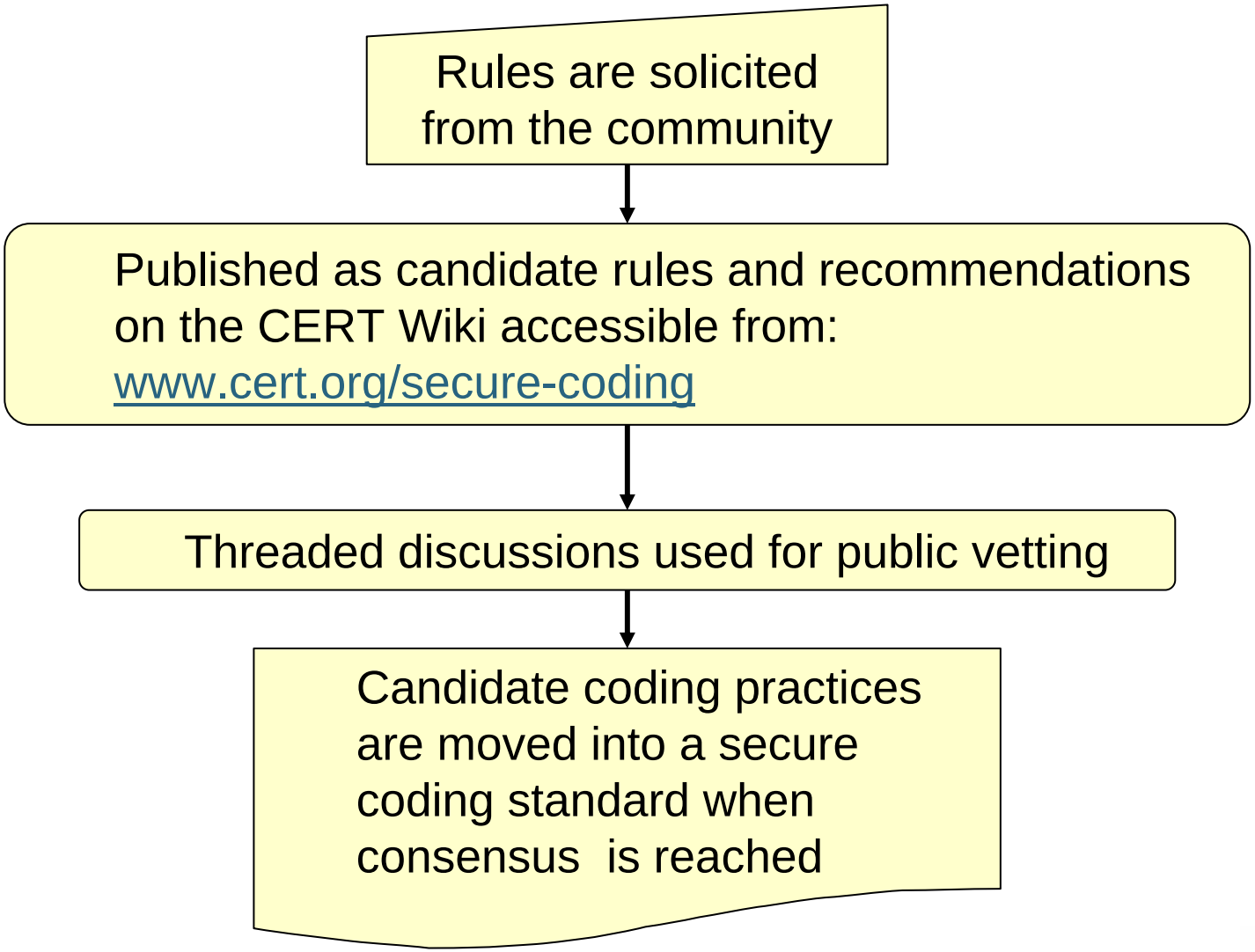
- Violation of the coding practice will result in a security flaw that may result in an exploitable vulnerability.
- There is an enumerable set of exceptional conditions (or no such conditions) where violating the coding practice is necessary to ensure the correct behavior for the program.
- Conformance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

Recommendations

Coding practices are defined as 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.

Community Development Process



Scope

The secure coding standards proposed by CERT are based on documented standard language versions as defined by official or *de facto* standards organizations.

Secure coding standards are under development for:

- C programming language (ISO/IEC 9899:1999)
- C++ programming language (ISO/IEC 14882-2003)

Applicable technical corrigenda and documented language extensions such as the ISO/IEC TR 24731 extensions to the C library are also included.

Potential Applications

Establish secure coding practices within an organization

- may be extended with organization-specific rules
- cannot replace or remove existing rules

Train software professionals

Certify programmers in secure coding

Establish base-line requirements for software analysis tools

Certify software systems

System Qualities

Security is one of many system qualities that must be considered in the selection and application of a coding standard.

System qualities with significant overlap

- Safety
- Reliability
- Availability

System qualities that influence security

- Maintainability
- Understandability

System qualities that make security harder

- Portability

System qualities that may conflict with security

- Performance
- Usability

Implementation & Demo

Externally accessible system hosted on the CERT web site

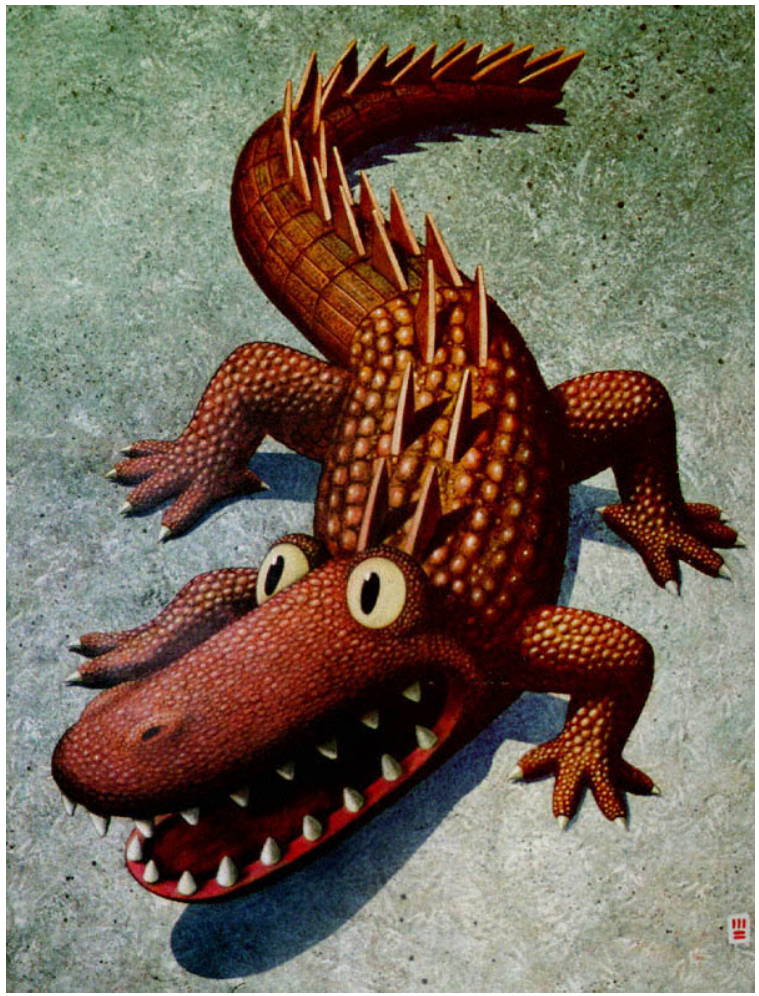
Software

- Atlassian's confluence wiki with unlimited named users

Hardware

- One Dell PowerEdge 2850
- Two Intel Xeon Processors at 3.0GHz/2MB Cache, 800MHz FSB
- Memory 2GB DDR2 400MHz (2X1GB
- Primary Controller Embedded RAID (ROMB)
- Three 73GB 10K RPM Ultra 320 SCSI Hard Drives

Demo



Future Directions

Provide similar products for other languages

- C++/CLI
- C#
- Java
- Ada
- Etc.

Produce language independent guidance cross-referenced with specific examples from target languages


Questions

For More Information

Visit the CERT® web site

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

Contact Presenter

Robert C. Seacord rcs@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**