Measuring the Cybersecurity Risk of Software-Intensive Systems

Marc Jones
Director, CISQ Federal Outreach

CISQ
International Standards for Automating Software Size and Structural Quality Measurement

Nine-Digit Glitches

Nine Digit Glitches

now affect

Board of Directors
CEO, COO, CFO
Business VPs
Corporate Auditors
CIO

accountable for

Governance
Risk management
Business Continuity
Brand protection
Customer experience

Evaluate Application Risk with CISQ Measures

© 2018 Consortium for IT Software Quality (CISQ) | www.it-cisq.org
Security Challenges in IoT Systems

- Broad attack surface with rapid propagation across components
- Components developed by different organizations
- Lack of shared cybersecurity information on component weaknesses
- Reliance on process certifications instead of software analysis

Modern Apps Are a Technology Stack

1. Unit Level
   - Code style & layout
   - Expression complexity
   - Code documentation
   - Class or program design
   - Basic coding standards
   - Developer level

2. Technology Level
   - Single language/technology layer
   - Intra-technology architecture
   - Intra-layer dependencies
   - Inter-program invocation
   - Security vulnerabilities
   - Development team level

3. System Level
   - Multiple languages
   - Architectural compliance
   - Risk propagation
   - Application security
   - Resiliency checks
   - Transaction integrity
   - Function points
   - Integration quality
   - Data access control
   - SDK versioning
   - Calibration across technologies
   - IT organization level
Skipping layers to access data can cause problems in:
- Security
- Data corruption
- Performance
- Maintainability

Detection requires analyzing transactions and data flows across languages and layers.
CISQ/OMG Standards Process

Automated Function Points
- Reliability
- Performance Efficiency
- Security
- Maintainability

OMG Approved Standards
- ISO Fasttrack
- Deployment Workshops

CISQ Executive Forums

CISQ Structural Quality Measures

<table>
<thead>
<tr>
<th>CISQ Structural Quality Measures</th>
<th>Example architectural and coding weaknesses included in the CISQ measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>22 weaknesses (Top 25 CWEs)</td>
</tr>
<tr>
<td></td>
<td>- SQL injection</td>
</tr>
<tr>
<td></td>
<td>- Cross-site scripting</td>
</tr>
<tr>
<td></td>
<td>- Buffer overflow</td>
</tr>
<tr>
<td>Reliability</td>
<td>29 weaknesses</td>
</tr>
<tr>
<td></td>
<td>- Empty exception block</td>
</tr>
<tr>
<td></td>
<td>- Unreleased resources</td>
</tr>
<tr>
<td></td>
<td>- Circular dependency</td>
</tr>
<tr>
<td>Performance Efficiency</td>
<td>15 weaknesses</td>
</tr>
<tr>
<td></td>
<td>- Expensive loop operation</td>
</tr>
<tr>
<td></td>
<td>- Un-indexed data access</td>
</tr>
<tr>
<td></td>
<td>- Unreleased memory</td>
</tr>
<tr>
<td>Maintainability</td>
<td>20 weaknesses</td>
</tr>
<tr>
<td></td>
<td>- Excessive coupling</td>
</tr>
<tr>
<td></td>
<td>- Dead code</td>
</tr>
<tr>
<td></td>
<td>- Hard-coded literals</td>
</tr>
</tbody>
</table>

An international team of experts selected the weaknesses to include in CISQ measures based on the severity of their impact on operational problems or cost of ownership.

Only weaknesses considered severe enough that they must be remediated were included in the CISQ measures.

CISQ Structural Quality measures are currently being extended to embedded systems software.
22 (of Top 25) CWEs Form the CISQ Security Measure

- CWE-22 Path Traversal Improper Input Neutralization
- CWE-78 OS Command Injection Improper Input Neutralization
- CWE-79 Cross-site Scripting Improper Input Neutralization
- CWE-89 SQL Injection Improper Input Neutralization
- CWE-120 Buffer Copy without Checking Size of Input
- CWE-129 Array Index Improper Input Neutralization
- CWE-134 Format String Improper Input Neutralization
- CWE-252 Unchecked Return Parameter of Control Element Accessing Resource
- CWE-327 Broken or Risky Cryptographic Algorithm Usage
- CWE-396 Declaration of Catch for Generic Exception
- CWE-397 Declaration of Throws for Generic Exception
- CWE-434 File Upload Improper Input Neutralization
- CWE-456 Storable and Member Data Element Missing Initialization
- CWE-606 Unchecked Input for Loop Condition
- CWE-667 Shared Resource Improper Locking
- CWE-672 Expired or Released Resource Usage
- CWE-681 Numeric Types Incorrect Conversion
- CWE-706 Name or Reference Resolution Improper Input Neutralization
- CWE-772 Missing Release of Resource after Effective Lifetime
- CWE-789 Uncontrolled Memory Allocation
- CWE-798 Hard-Coded Credentials Usage for Remote Authentication
- CWE-835 Loop with Unreachable Exit Condition ('Infinite Loop')

Update to CISQ measures:
- Extensions for embedded
- Additional critical weaknesses
- Expected 2H 2019
- CWE Parent-child structure:
  - 34 parents
  - 41 children

CISQ and the NIST Cybersecurity Framework

The CISQ Security measure (and others) can be used in numerous processes of the NIST Cybersecurity Framework. Some examples:

- Empirical risk tolerance thresholds for software security
- Contractual SLAs and audits for software security
- Evaluation of software assets for security weaknesses
- Continual improvement of software security
- Periodic scans for software weaknesses
- Software security and weakness data are shared
- Security weaknesses are identified and mitigated

The CISQ structural quality measures play an important requirements and verification role for ‘Build Security In’ approaches to cybersecurity.
CISQ Conforms/Supplements ISO 25000 standards

- ISO/IEC 25010 defines a software product quality model of 8 quality characteristics
- CISQ conforms to ISO/IEC 25010 quality characteristic definitions
- ISO/IEC 25023 defines measures, but not automatable or at the source code level
- CISQ supplements ISO/IEC 25023 with automatable source code level measures

CISQ automated structural quality measures are highlighted in blue

CISQ-like Measures Predict Incidents & Costs

\[ R^2 = .34 \]
Total Quality Index accounts for 1/3 of variation in incidents

Increase in Total Quality Index of .24 decreased corrective maintenance effort 50%
Application Certification Using CISQ

- CISQ/OMG
  - only assess vendor conformance
  - do not certify applications
- Service providers
  - use CISQ-conformant technology
  - in a CISQ-conformant service process
  - to provide application certifications

Deploying CISQ Measures

- CISQ measures
- OMG standards
- ISO standards
- Federal IT Policy
- Corporate IT Policy
- Regulations
  - Sec. & Exch. Com. Sec.
  - State of Texas
- System acquisition
- Third party Contracts
- Benchmarks
As a greater portion of mission, business, and safety critical functionality is committed to software-intensive systems, these systems become one of, if not the largest source of risk to enterprises and their customers. Since corporate executives are ultimately responsible for managing this risk, we establish the following principles to govern system development and deployment.

1. **Engineering discipline in product and process**
2. **Quality assurance to risk tolerance thresholds**
3. **Traceable properties of system components**
4. **Proactive defense of the system and its data**
5. **Resilient and safe operations**