Security Metrics – The Latest From Metricon 2.0 &…

Presented To:
Central Virginia ISSA
September 12, 2007

Presented by Bob Austin
austinb@korelogic.com
804.379.4656
Disclaimer

• Some of the information presented herein is taken from presentations at Metricon 2.0 or from my notes. Sources are credited wherever possible.
Background

• securitymetrics.org
  – A community website for security practitioners.
  – Metrics Blog
    (http://www.securitymetrics.org/content/Wiki.jsp?page=Welcome)

• Metricon 2.0:
  – An annual workshop on security metrics, August 7th, Boston
  – Metrics thought leaders: Andrew Jaquith, Dan Geer, Adam Shostack, etc.
Do Metric Matter?

Andrew Jaquith

- Security~the “Seinfeld” of IT
- Progress over time
- “Happy metrics”
- Build bridges to those who speak numbers
- Compare LOBs, cross sectional comparisons
- “You are here” charts
- Benchmark against industry peers
- Make people curious – “if your numbers are boring, you’re using the wrong numbers.” Tufte.
- Compliance frameworks
  - Useful to start a security program
  - Not to make business risk decisions
  - “Security Balanced Scorecard” > compliance framework

Mike Rothman

- Tons of operational metrics but few are relevant to the folks that run your business.
- Use the language of business…align metrics with the business
- You need to figure out what is going to resonate with your management team, and the only way to do that is to talk to them. Remember, your desk is a dangerous place to spend your day.
Security Metric System to Rate Enterprise Software (Fortify)

• Reviews Open Source Java projects for
  – Security defects
  – Quality defects
• [website](http://opensource.fortifysoftware.com/)
• 74+ millions of lines scanned weekly
• Over 130+ Open Source Projects
• Users immediately asked “**which is better?**”
  – Which project is more secure?
• But……..
  – Analyzing the source code only uncovers vulnerabilities
  – Risk information should be derived by the consumer of the software
  – Lacking complete information, we steer clear of risk determinations
  – We can’t answer:
    • “How secure am I?”
    • “Where should I place my security resources?”

* Source: Fortify
Categorizing Applications (Fortify)

- Applications are categorized to aid in “apples to apples” comparisons of projects
  - “Show me the best mid-sized CRM system”
- Genres are a combination of:
  - Application Size
    - Small < 150KLOC
    - Medium > 150KLOC; < 500KLOC
    - Large > 500KLOC
  - Application Type
    - Web Application
    - Application Server
    - Component/API
    - Stand-alone utility
    - Console application
  - Application Industry
    - Financial
    - Engineering
    - Consumer
    - eCommerce
Rating System (Fortify)

• Modeled a “star” system similar to mutual fund ratings
  – Morningstar for Software Security
• Developing a 5-star/tier rating system
  – Some subtlety of assessment is lost in exchange for ease of use
• Each tier has criteria before a project may be promoted to the next tier
• NOT limited to static analysis tools
Stars Explained (Fortify)

- Absence of Remote and/or Setuid Vulnerabilities
- Absence of Obvious Reliability Issues
- Follow Best Practices
- Documented Secure Development Process
- Passed Independent Security Review
Critique of Rating System (Fortify)

• Ouch!
  – Ratings are very harsh
    • Impact is high for even one exploit
    • If automated tools can find issues, other security defects likely to exist

• Ordering in unordered set
  – The tiered nature implies more importance for some criteria
  – Project information is loss
    • A project may have a defined security process, yet have vulnerabilities present

• Subjectivity of higher tiers
  – Things become more ambiguous as you move up the tiers
    • What qualifies as an independent review?
Using the System (Fortify)

• Search for software that meets a particular security requirement
  – “Show me 2-star, mid-size, shopping cart software”
• Use stars to filter out components
• Drill down to underlying metrics to make a more informed decision on component usage
  – “How does this set of 1-star components compare?”
    • Examine detailed defect information
    • Does one project have fewer remote exploits?
• Feed metrics into existing risk assessment process to make final determination
  – Model how the selected software will impact existing risk model
Next Steps? (Fortify)

• Validate against Closed Source software
• Test against auditor “smell test”
  – How do the hard numbers compare against security auditors reports?
• Explore additional software metrics that may relate to security
Security Metrics Panel

• GE, Wellpoint, Booz Allen
• Lessons learned on building GE’s metrics program:
  – Like all security initiatives, start with a executive champion.
  – Create a cross-functional metrics “steering” committee (key stakeholders) – what do they need for decision-making?:
  – Need a metrics “marketing” plan to communicate the business benefits.
  – The LOB “horse race” factor – no one wants to be last or even below the mean. Use this as a lever for change.
Security Metrics Panel

• What metrics?
  – In the beginning used the wrong metrics (# of patches, etc.)
  – Had to move to “so what” metrics:
    – What are the risks that I can do something about?
    – What are risks associated with Project X? For example, Project X will sharply increase the # of 3rd parties who handle our data. What the risk and how do we monitor it (metrics)?
  – Six Sigma-based: opportunities and defects
  – 2 types of metrics used: compliance/policy, operational
  – Better to have no metrics than bad ones.
  – Trend accuracy > point precision.* “Directionally correct” is the best you can do in some cases and will support decision-making

* Dan Geer
Security Metrics Panel

- **Automate!**
  - More likely to produce repeatable (therefore trusted) results.
  - Cheaper than manual collection.
  - ClearPoint Metrics – a metrics dashboard development framework. GE has had good success with the product and the company.
  - Metrics delivered via dashboard portal = convenient.
  - Report type examples:
    - “Monolithic” corporate metrics (e.g., centralized scans)
    - Trend charts
    - Comparative views (e.g., comparing LOBs)
Security Metrics Panel

- Who are you communicating with?
- What is your message/story?
- Why is your message important?
- Audience preferences regarding reports
- Execs want bottom line information

- Technical versus management
- Metrics aren’t the story; they support your story.
- “So what” factor. Tailor to what they need.
- Level of detail desired varies among execs
- What is your conclusion? What decision needs to be made?

EXAMPLE - The right metric for the right audience:
- Technical: 500 firewall rule changes, 2 failures = 99.6% success rate
- Mgmt: Business impact: 2 failed firewall rule changes created a 1 hour outage and exposed PHI data
Security Metrics Panel

- Figuring out what metrics execs want…..ask them:
  - What things concern you that would negatively impact our financial performance, brand, etc?
  - “What are the risks I can do something about?”
    ➔ Translate this into how the security program can support it (e.g., how to protect brand) and other corporate priorities.
    ➔ Create metrics to support how the security program’s contributions

- Key risk questions execs are interested in?
  ➔ Are we “safe”?
  ➔ Are things under control?
  ➔ Is our critical data well protected?
  ➔ Compliance – keep us out of jail?

- **Time** metrics: operational efficiency, reduced cycle time
- **Financial** metrics: increased productivity, lower costs, lower headcount
Security Metrics Panel

• Don’t use metrics to:
  – As the **sole** argument for funding. Metrics aren’t the story; they support your story. They should support your business case
  – Tell management how “big the problem is”

• “Telling your CEO how long it takes you to patch a server or what percentage of AV coverage you have has really helped your standing in the organization, you should keep doing that. I hear that'll put you on the fast track for bigger and better things. Yes, I'm joking.”*

* Quote from the Pragmatic CSO, Mike Rothman
C-level Metrics

- Model in alignment with your CIO’s metrics –
  - The CIO (presumably the boss of the CISO) has a set of metrics that are tracked because they are important to the business.
  - Take those metrics as a starting point and derive a set of security metrics accordingly.

- C-level metric candidates:
  - Uptime, availability
  - Intellectual property protection
  - Corporate liability
  - Compliance
  - Corporate brand

- A potential metric - "Security Value Destruction."
  - Very hard to show security ROI or adding to the bottom line
  - But, security - if done poorly - can cost the business a boatload.
  - Security stupidity cost the organization - track that and hopefully show improvement.
  - Count things:
    - Cost of downtime,
    - Legal fees to fight off law suits directly related to policies being violated,
    - Audit penalties
    - Market cap decline after a publicly announced breach.
    - Goal is to you can keep it at roughly ZERO over time.
    - Having a bad year relative to Security Value Destruction doesn't mean you suck at security. It means you got nailed. Join the club.

Source: Pragmatic CSO, Mike Rothman
My Takeaway from Metricon

• Interest in metrics continues to build; people are starting to tackle harder measurement issues.
• Lots of head nodding regarding metrics (especially C-level), but few concrete examples offered.
• Consistent theme: Start small; build on early successes.
More Metrics
# Metrics: One Size Does Not Fit

<table>
<thead>
<tr>
<th>Audience</th>
<th>Metrics Needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Senior management</td>
<td>Confidence in external reporting of regulatory compliance, enterprise risk management status, How do we compare to peers? What risks exist that would impact our brand, bottom line?</td>
</tr>
<tr>
<td>Business Unit Mgmt</td>
<td>Understand risk posture of business unit. Understand status of security obligation to business partners. What risks exist that would impact our business unit brand, bottom line?</td>
</tr>
<tr>
<td>CIO</td>
<td>Identify/address systematic problems and flaws in development and system operational processes that increase risk. Is risk increasing or decreasing over time?</td>
</tr>
<tr>
<td>CISO</td>
<td>Inform and advise senior management in managing risk. Manage and improve security processes.</td>
</tr>
<tr>
<td>System or Application Owner</td>
<td>Security status <strong>before</strong> and after they take over the operations from development.</td>
</tr>
<tr>
<td>IT Staff, Developers</td>
<td>To document security status. Use to improve performance.</td>
</tr>
</tbody>
</table>
“Security metrics must be more tightly linked to the business and communicated in simple terms.*”

Particle Physics for a Physicist**
(=Security Engineer)

- String theory
- Fermions, neutrinos
- Hadrons, quarks, antiquarks
- Strangelets
- Higgs Field

- Security equivalent = “SQL injection vulnerabilities are present in……an attacker can cause malicious SQL code to run by maliciously modifying data used to compose an SQL command…..”

Particle Physics for the Manager**
(= Application Owner)

- What is this table made of?
- How do the particles in it constitute a “table”?

- Security equivalent = “This vulnerability may allow an attacker to harvest customer data.”

* Source: Embedding Information Security Risk Management into the Extended Enterprise, Center for Digital Strategies, Tuck School of Business at Dartmouth

** Source: Jos Engelen, CERN Chief Scientific Officer, quoted in The New Yorker Magazine, May 14, 2007
Snapshot of the OWASP Survey

Survey Sections
• Participant and Application Portfolio Profile
• Application Security Metrics Program Profile
• Application Security Metrics Tools Profile

Application Security Metric Description
• Name of Metric
• Description of Metric
• Formula or description of how metric is created?
• Source of the data used to produce the metric?
• How is the metric data collected?
• How is the metric used?
• Who is the primary recipient of this metric?
• How frequently is the metric produced, reported?
• How are metrics delivered to consumers?
• Effort required to produce/report the metric
Web App Metric Survey Response (The “Mean”)

• Name of Metric
• Description of Metric
• Formula or description of how metric is created?
• Source of the data used to produce the metric?
• How is the metric data collected?
• How is the metric used by recipients?
• Who is the primary recipient of this metric?
• How frequently is the metric produced, reported?
• How are metrics delivered to consumers?
• Effort required to produce/report the metric

→ Security Test Findings
→ Vulnerabilities
→ Automated testing tool output and manual testing results
→ Automated testing tool output and manual testing results
→ Automated and manual testing
→ Evaluate high risk application security flaws
→ Application Security or Information Security teams
→ Ad hoc, as needed or requested
→ Reports are emailed to recipients
→ 6 – 10 hours
Metrics from Vendors

• Initial Surveyed*:
  – Fortify – source code analyzer
  – Watchfire, SPI Dynamics – web application vulnerability scanner
  – Archer - risk and compliance management information management framework

• Areas of inquiry:
  – Current metrics and reports provided by your product
  – How do your customers use the metrics/reports besides for vulnerability mitigation?
  – What (new metrics) are your customers asking for (developers, security, mgmt, audit)?
  – To the extent that you are able, what is your metrics vision for the product over time?
  – Maturity of application security metrics programs among your clients--what is the “mean”?
  – Which of these metrics/reports do your customers value most?

* Important caveat – This is not a product endorsement of these vendors.
Measurements/Metrics (Fortify Software)

Measurements/Metrics

• Lines of Code (LOC) scanned. % total LOC scanned.*
• Lines of Code scanned/unit time.*
• #/% of applications scanned.

• #, % of application development projects run through security gate checks
• #, % of application development projects run through security gate checks which failed.

Fortify’s Explanation of Metric

➔ Relative measure of gap/risk in untested code base.
➔ Testing “efficiency” or testing “capacity” (e.g., 2M LOC more in 2007)
➔ Relative measure of tested/untested applications. Relative compliance metric.
➔ Higher confidence level in SDL-produced apps. Should show correlation to decreasing # of vulns at SDL gate.
➔ Applications released with flaws (exemptions granted)

* Obviously, these do not speak to the accuracy of the testing.
Measurements/Metrics (Fortify Software)

Measurements/Metrics

- Vulnerability Severity Comparison between Projects
- Vulnerability Category Comparison between Projects
- Vulnerability “dwell” by vuln severity
- Vulnerability Category-specific trending

Fortify’s Explanation of Metric

- Relative comparison of “like” applications. Project efficacy.
- May highlight flawed development in a given category (SQL injection)
- Avg time to mitigate (by severity level)
- Some clients focus on avoiding certain vulnerabilities

* Obviously, these do not speak to the accuracy of the testing.
### OWASP 2004 Top 10/PCI Section 6.5 Compliance

#### Project Information
- **Project:** Fairbanks
- **Description:** Even numbered scans.
- **Last Scan Date:** 11/12/2005

#### PCI Requirement 6.5:
Develop Web software and applications based on secure coding guidelines such as the Open Web Application Security Project (OWASP) guidelines, review custom application code to identify coding vulnerabilities and over prevention of common coding vulnerabilities in software development processes, to include:
- Unvalidated input
- Broken access control
- Broken authentication/session management
- Cross site scripting attacks
- Buffer overflows injection flaws (e.g., SQL
- Improper error handling
- Insecure storage
- Denial of service
- Insecure configuration management

#### OWASP 2004 Top 10/PCI Section 6.5 Compliance Category

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Hot</th>
<th>Warning</th>
<th>Info</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1. Unvalidated Input</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>A2. Broken Access Control</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A3. Broken Authentication and Session Management</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A4. Cross-Site Scripting (XSS) Flaws</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>A5. Buffer Overflows</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A6. Injection Flaws</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>A7. Improper Error Handling</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>A8. Insecure Storage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A9. Application Denial of Service</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>A10. Insecure Configuration Management</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

**Total:** 7, 41, 10, 58
Compliance-based Metrics (Archer)
Vendors – Metrics Opportunity

• Useful regulatory compliance reporting (esp Watchfire and Archer).
• Many of the metrics/reports provided by the vendors are from client requests.
• Win-Win: Automated tool vendors have the opportunity to differentiate themselves with metrics that meet clients’ needs.
Web Application Security Metrics Framework
Application Security Metrics Framework

Overview

• Excellent approach!
• Gunnar Peterson and Elizabeth Nichols
• Published in *IEEE Security and Privacy Magazine*, March/April, 2007

• Framework Objective:
  – Propose metrics to help quantify the impact that process changes in one lifecycle phase has on other phases.
  – Map metrics to the OWASP Top 10
  – Create a security scorecard

• Currently used by several clients of Peterson and Nichols to track secure coding improvements

• The Framework was developed independently of the OWASP Web Application Security Metrics Project
Application Security Metrics Framework

Design Phase

- Use:
  Make risk mgmt decisions during design phase.

- Benefit:
  Identify flaws early in the SDL.

Example
Unvalidated input (% Validated Input)

Deployment Phase

Use:
Measure changes to the target system over time

Benefit:
Provide insight into rate of change, SLA metrics (availability)

Example:
Insecure configuration mgmt (# of service accounts)

Run-time Phase

Use:
Web app behavior in production & vulns found after deployment

Benefit:
Ideally, metrics from previous phases show improvement that result in fewer run-time vulnerabilities

Example:
 Broken Authentication & Session Mgmt (Broken Account Count)

Source: “A Metrics Framework to Drive Application Security Improvement”, IEEE, Elizabeth Nichols and Gunnar Peterson
### Example of a Framework Metric

<table>
<thead>
<tr>
<th>OWASP Type</th>
<th>Metric Name</th>
<th>Description/ Formula</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unvalidated Input</td>
<td>“%-Validated-input”</td>
<td>= V/T where V = # of interfaces that use input validation mechanisms T = # of input forms, interfaces the application exposes (# of HTML Form POSTs, GETs, etc.)</td>
<td>High % = more input validation mechanisms in place</td>
</tr>
<tr>
<td></td>
<td></td>
<td>How vulnerable is the web app to input validation attacks?</td>
<td>Low % = fewer input validation mechanisms in place</td>
</tr>
</tbody>
</table>

**Application Security Program Implications, Root Cause Analysis:**
- If most of a company’s web apps have low % scores, implies:
  - Developer secure coding training deficiency?
  - Missing or inadequate secure application standard?
  - Trend monitoring for improvement over time

Source: “A Metrics Framework to Drive Application Security Improvement, IEEE, Elizabeth Nichols and Gunnar Peterson”
### Example of a Framework Metric

<table>
<thead>
<tr>
<th>OWASP Type</th>
<th>Metric Name</th>
<th>Description/ Formula</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insecure configuration mgmt</td>
<td>“# of service accounts”&lt;br&gt;A secure application requires a secure server.</td>
<td>Determine number of weak or default passwords</td>
<td>High # = relative measure of risk of unauthorized access.</td>
</tr>
</tbody>
</table>

Source: “A Metrics Framework to Drive Application Security Improvement, IEEE, Elizabeth Nichols and Gunnar Peterson”
## Framework: Run-time Metrics

<table>
<thead>
<tr>
<th>OWASP Type</th>
<th>Metric Name</th>
<th>Description/Formula</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-site scripting (XSS)</td>
<td>“XSSVulnCount”</td>
<td># of XSS vulnerabilities identified by testing</td>
<td>High # = relative measure of risk of vulnerability to XSS attacks</td>
</tr>
</tbody>
</table>

Source: “A Metrics Framework to Drive Application Security Improvement, IEEE, Elizabeth Nichols and Gunnar Peterson”
Other Framework Metrics

**OWASP Category**
- Broken Access Control
- Broken Authentication and Session Mgmt
- Buffer Overflow
- Injection Flaws
- Improper Error Handling
- Insecure Storage

**Framework Metrics**
- Anomalous Session Count
- Broken Account Count
- Overflow Vulnerability Count
- Injection Flaw Count
- Instance/Application Count
- % of servers with automatic and HDD encryption

Source: “A Metrics Framework to Drive Application Security Improvement, IEEE, Elizabeth Nichols and Gunnar Peterson
**Objective**
The objective of the scorecard is to provide an overall status on application security organized by OWASP Top 10 categories.

<table>
<thead>
<tr>
<th>OWASP Category</th>
<th>Rating</th>
<th>Trend</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unvalidated Input</td>
<td>94</td>
<td>↑</td>
<td>Design Standards now in place</td>
</tr>
<tr>
<td>XSS</td>
<td>89</td>
<td>⇧</td>
<td>Results from pen test this month</td>
</tr>
<tr>
<td>Buffer Overflow</td>
<td>79</td>
<td>⇩</td>
<td>Results from pen test this month</td>
</tr>
<tr>
<td>Overall Mean</td>
<td>87</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- = +91
- = 81-90
- = <80

Redrawn, in part, from “A Metrics Framework to Drive Application Security Improvement”
A Metrics Program Start-up Checklist

- Read Security Metrics by Andrew Jaquith
- Be able to articulately answer the following question: “We are implementing a security metrics program because….”
- Adhere to measurement best practices:
  - Measure what?, Why measure it? Measure it for whom?
- Interview key stakeholders to determine what they want
  - Refine the stakeholders information needs into measurable items
- Start with a manageable, useful set of metrics. Use these filters:
  - Is it valuable to the organization?
  - Are the metrics valid internally? Externally?
  - Is the metric reliable?:
  - Can it be used to improve shortcomings?
  - How hard is it to produce (automation wherever possible)?
- Have a good estimate of the time to produce each metric.
- Don’t forget to set a baseline. A ”must have” to show improvement.
- Consider including:
  - Normal range
  - Monitor range
  - Replan range
Thanks for Listening

Q&A
Resources – Security Metrics

- securitymetrics.org
- Metrics Blog
  (http://www.securitymetrics.org/content/Wiki.jsp?page=Welcome)
- Metricon 2.0 presentations:
  http://www.securitymetrics.org/content/Wiki.jsp?page=Metricon2.0
- Security Metrics: Replacing Fear, Uncertainty, and Doubt, Andrew Jaquith
- Dan Geer’s “Measuring Security Tutorial”
- Data visualization by Edward Tufte
  – Beautiful Evidence
  – The Visual Display of Quantitative Information, 2nd Edition
  – Envisioning Information
- www.clearpointmetrics.com – a metrics dashboard development framework product, the current market leader