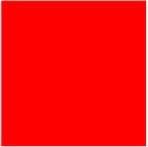


ORACLE[®]

The Good, The Bad, And The Ugly: Stepping on the Security Scale

Mary Ann Davidson, Chief Security Officer



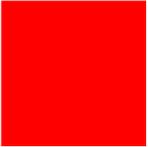
How I Became a Metrics Fiend

- Positives
 - Rudy Giuliani's *Leadership*
 - Knowledge is power: development will manage their own issues *if they know what they are*
- Negatives
 - PR response fire drills
 - Endless discussions on software assurance/cybersecurity – measurement - many led by those with no actual business experience
 - *Nui ka* acquisitions
- Conclusion: Needed simple, fast, accurate way to know “how we are doing?” and “where are we?”



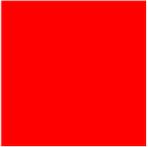
Agenda

- What Makes a Good Metric
- Private vs. Public Metrics
- Potential Security Metrics
- Triangulated Metrics
- Metrics Portal
- Governance
- Conclusions



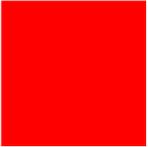
What Makes a Good Metric?

- Should help you manage better (not merely assign blame)
- Should motivate good/correct behavior (not promote evasive tactics just to make the numbers look good)
- Should prompt additional questions (“Why? How?”) to understand what is influencing the numbers
- Should help answer basic questions of goodness (e.g., “Are we doing better or worse?”)
- Should be objective and measurable, even if correlation may not equal causality
- Should (in some cases) include “triangulation” so you can fix your position (e.g., latitude AND longitude needed)
- Note: “Make you look good to third parties” is **not** on the list



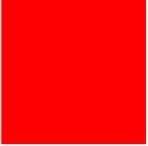
An Approach to Measurement

- What information do you already have?
 - Data mine what exists!
- What would you like to know?
- What is the value (and cost) of getting more information?
 - If value < cost, skip it!
- What factors influence the metric, including shortfalls of data sources?
- Resource metrics as you would any *other* business function
- “It does not have to be perfect to be useful and something is usually better than nothing.”



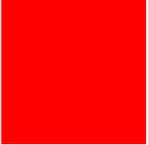
Public vs. Private Metrics

- Generally private data is far more honest
 - Ask a stranger his/her age, weight, golf score...
 - The scale doesn't lie...
- The mere fact of publishing incents cheating (Heisenberg's Uncertainty Principle)
 - Airbrushing is not just for photos
 - Climategate
 - Nobody will publish numbers showing they are getting suckier ... *and* they want their competitors to look suckier than they are
- Transparency is a good thing...in time
 - "Before and after" pictures are more successful than "before" pictures



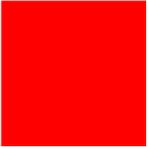
Example – Oracle Global Product Security (GPS)

- Responsible for security program management across all Oracle
- Focus areas include:
 - Assurance – engineering security into development
 - Product assessments/ethical hacking
 - Security evaluations
 - Security vulnerability handling
 - Secure development compliance
- Primary focus of our metrics program is assurance and vulnerability handling
 - Potential cost savings for us and for customers in improving assurance and vulnerability handling



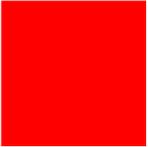
Potential Basic Security Metrics (1)

- Defects per KLOC
 - + objective, measurable, “standard,” “like to like” comparison
 - +- primarily a quality metric
 - - says nothing about severity, exploitability
- Number of publicly announced vulnerabilities
 - + could be a rough “security comparison” metric
 - - does not factor complexity of code, size of code base, how product is factored
 - - unless vendor publishes their disclosure policy, metric is “rigged to cheat”
 - - does not facilitate apples to apples comparison because there is no way to normalize the data



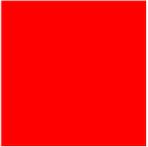
Potential Basic Security Metrics (2)

- “Time to fix” security vulnerability
 - Should compute average, by team, by reporter, etc.
 - + motivates attention to critical issues – fix fast
 - - “complete” fix is more critical than “fast” – security bugs should never be “reopened”
 - - “fix” hard to measure consistently (Base code? Next patch? All old versions?)
- Top five most common vulnerabilities by development group
 - + helps find problem areas to focus resources, including training, tools, “extra attention,” ethical hacks...



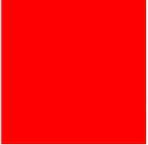
Potential Basic Security Metrics (3)

- Who finds the most vulnerabilities? (security researchers, internal, customer)?
 - + Good motivational metric, good secondary uses
 - +- Trend is more important than absolute numbers
 - +- Helps spot “targets of opportunity”
- Code coverage and “usage” of automated security vulnerability finding tools
 - + Consistency and breadth of coverage
 - - No automated tool finds everything; they all have different strengths



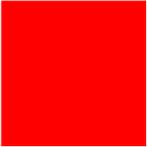
Data Source Challenges – Examples

- Hard to determine “exploitability” of bugs in all cases (e.g., some bugs may be exploitable only by the administrator - who can do everything, anyway)
- Vulnerabilities found by automated tools are generally fixed without bugs being logged (so that data cannot be mined)
- How to count bugs in beta versions (that are fixed *before* production?)
- Many bugs filed as part of automated scans report many problems in one bug (other logging is “one problem, one bug”)
- Multiple bug repositories



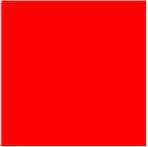
Other Security Metrics

- Basic secure coding training statistics
 - Class required by (virtually all) development, up to and including SVPs
 - By development team, who has attempted secure coding class?
 - By development team, who has completed and passed secure coding class?
 - Metric reported to senior management and internal audit
- Advance training class statistics (against target group)
- While “correlation does not equal causality,” the above are “good hygiene metrics,” especially in dynamic organizations



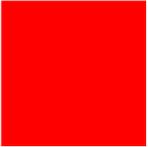
Combined Metrics - Background

- Oracle issues quarterly security patches called critical patch updates (CPUs)
 - Bundled, high severity security issues
 - <Generally> found by researchers
 - Dates announced a year in advance
 - Patches issued for all supported versions on all affected operating systems
 - Testing done with dependent products (and *their* CPUs)
 - (Most) products are in the CPU program



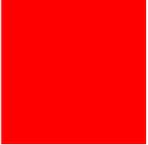
A Triangulated Metric Example (Part 1)

- Percentage of security patches completed by published deadline
- Percentage of security patches that have to be reloaded after publication
- Customer service requests against downloaded security patches
- Number of security vulnerabilities that drop out of patches as they go *through* the patch process
- Total number of fixes delivered in a patch
- Overall: Goal is to deliver patches on time, with high quality, fixing identified “critical” security issues



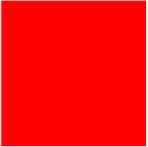
A Triangulated Metric Example (Part 2)

- Third party (security vendor) survey alleged
 - Only 1 of 10 DBAs regularly applies Oracle Critical Patch Updates
 - Two-thirds have never installed CPUs
- Issue with metric:
 - Customers *also* obtain security fixes through patch sets and upgrades
 - CPUs are cumulative for most products (ergo, you do not have to apply every single one ...)
 - Most customers do not run monolithic (all on same product version) enterprises
 - Most customers do not apply all patches from all vendors
 - Customers answered the question that was asked, not the one that wasn't asked
 - And...we did our own survey and it did not concur with the third party survey



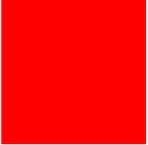
Vulnerability Metrics Portal

- Aged backlog of security vulnerabilities and aging trends
- Supports drill down to individual vulnerabilities
- Slicing and dicing of vulnerability information
 - By product
 - By “who found” (internal, customer, external researcher...)
 - Or “all”
- Enhanced per development feedback
- Development has access so they manage their own backlog



Story Time: The Good (1)

- Blog entry on “responsible disclosure/responsible remediation” (<http://blog.osvdb.org/2009/11/15/responsible-disclosure-old-debate-fresh-aspects>):
 - Vendors should disclose all vulnerabilities fixed in new versions (but not necessarily patch in old ones) on grounds that customers will get a benefit if they learn about internally discovered and fixed bugs and will be more aggressive in upgrading to new version (Unproven)
 - Posits if one researcher finds a vulnerability then another will, so there is no downside of disclosing internally found bugs: discovery is “inevitable” (Unproven)
- Our analysis for a product also focused on by researchers:
 - Researchers find 3% of vulnerabilities
 - Customers find 10% of vulnerabilities
 - Internally we find 87% of vulnerabilities
 - And, less than 1% of product vulnerabilities found internally are *also* found by researchers



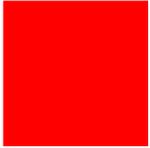
Story Time: The Good (2)

- Questions raised by analysis
 - If we took the resources that were used to discover the 86% of the bugs found internally but not by researchers and focused those resources on the 2% of the bugs found by researchers and not found internally...
 - **And** were able to find and fix all the vulnerabilities that the researchers would find as a result...
 - We might “get ahead” of the Oracle focused researcher pool



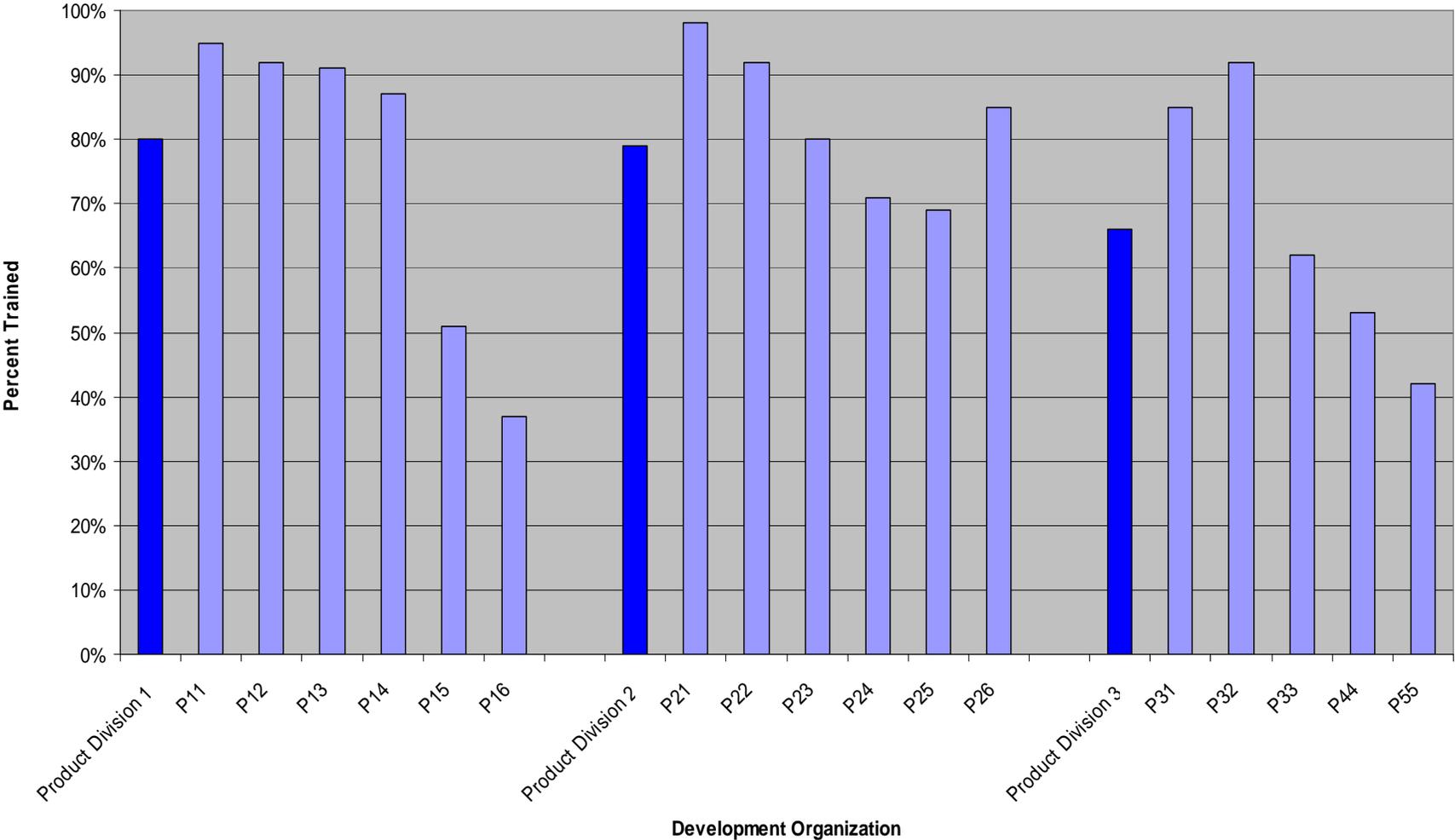
Story Time: The Bad

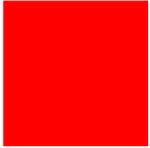
- Despite “mandatory” online secure coding training (on Oracle Secure Coding Standards), many development groups lagged in compliance
 - “We’re in the middle of code freeze...”
 - New hires/transfers
 - New (acquired) lines of business
- GPS reported new training numbers to Oracle Security Oversight Committee (OSOC) and CEO
 - ...and announced this two months *in advance* to development and encouraged them to “get numbers up”



Training Reported to OSOC

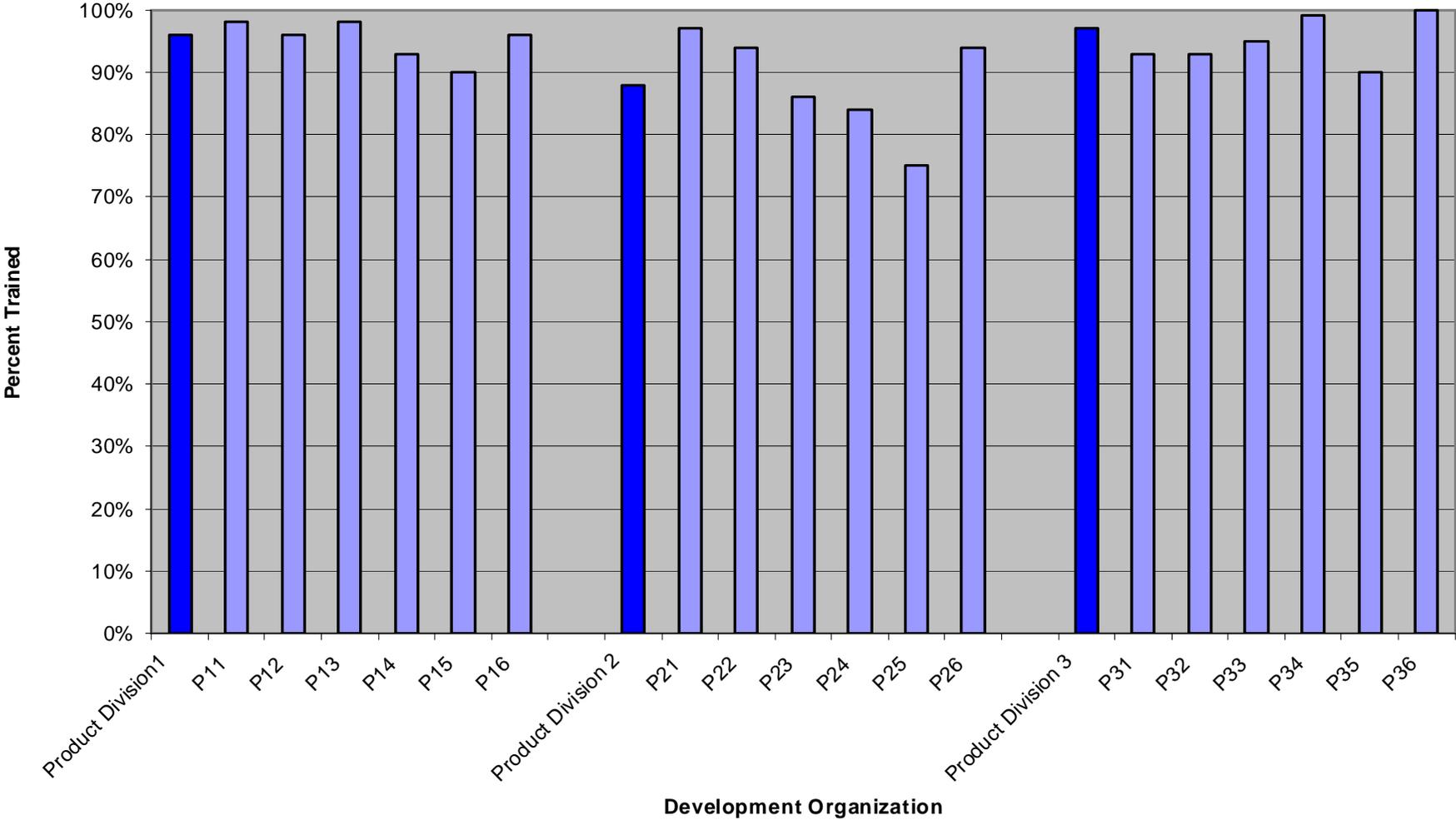
Secure Coding Practices Completion Summary

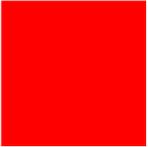




Training Four Months Later

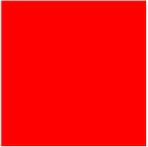
Secure Coding Practices Completion Summary





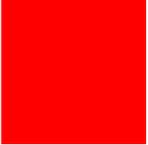
Story Time: The Ugly

- Customer expressed reservations about the security of an Oracle product
 - Customer had already experienced IP exfiltration (not related to Oracle)
 - Customer was in security sensitive industry
- Oracle considered providing a written description of assurance measures in interests of transparency
 - Based directly on compliance spreadsheet but not showing “compliance/non-compliance” or other product comparisons
- Conclusion: Product group is now officially a “problem child”:
 - “Cure plan” has been reported to EVP of product division
 - “Fix” will be reported to and tracked by Oracle Security Oversight Committee
 - Using “customer concern” as a “taser” for change



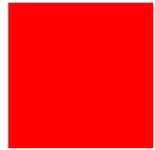
Story Time: They Lived Happily Ever After

- Supply chain issues/concerns/questions on the horizon of many customers (and regulators)
- Oracle already has a separate source code project in process with goals:
 - Protect our IP
 - Be able to tell our story regarding supply chain risk – across each line of business
 - Have a voice at the public policy table as supply chain risk is discussed
- Eureka moment:
 - Security scorecard/governance lends itself to supply chain compliance
 - Can extend existing assurance scorecards for supply chain risk
- We can leverage existing metrics framework for another (related) purpose more easily than would be the case without it



Conclusions/Recommendations

- “Responsibility without authority = frustration.”
- “There are lies, damn lies, and statistics.”
- “Start measuring somewhere.”
- “Don’t assume malice or incompetence if there is another explanation.”
- “Manage with metrics, not *to* them.”

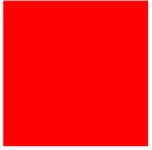


For More Information

- www.securitymetrics.org
- Dan Geer, currently of In-Q-Tel
(http://en.wikipedia.org/wiki/Dan_Geer)
- *Leadership* by Rudy Giuliani



ORACLE IS THE INFORMATION COMPANY



ORA