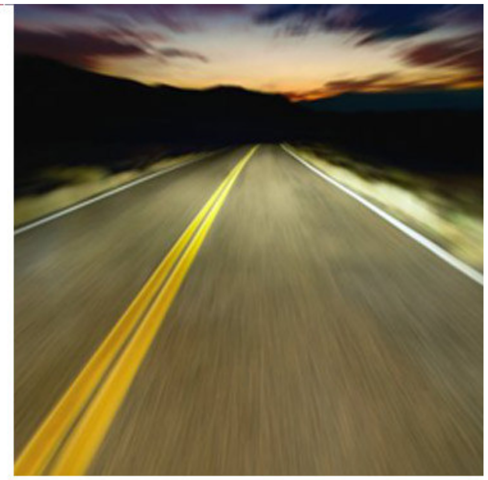
INTEGRATE. ASSEMBLE. OPTIMIZE.



“Good Enough” Metrics

Jeremy Epstein
Senior Director, Product Security
webMethods, Inc.

webMethods. Get There Faster.™

The problem…

We’re spending all our time arguing about *which* statistics we should gather

Instead, we should gather *all* the numbers we can, and then figure out which ones matter

Some metrics we can gather today

Lines of code

Language(s) used

Complexity metrics

Distant relationship to vulnerabilities

# of CVE entries reported

# of Bugtraq reports

# of vendor patches

Retrospective

# of problems found by scanners

# of problems found by fuzzers

# of problems found by static analyzers

Tendency to large fraction false positives; no standardization

Relative Vulnerability – a real metric
(kudos to Crispin Cowan, Novell)

Concept: Given a product and some # of exploitable vulnerabilities in the product, measure % exploitable with and without an intrusion prevention system (IPS)

Hypothesis: the IPS-protected version should have consistently fewer vulnerabilities than the product itself (and not introduce new vulnerabilities)

Tested with Immunix vs. Red Hat 7.0 & 7.3

Questions

- How to extend to applications where there is no IPS?
- Is a metric for IPSs really what we want?

Leading Security Indicators – A Real Metric
(aka "Seven Deadly Sins")

Goal: Estimate how good or bad software is likely to be by the self-reported answers to a handful of key questions

Hypothesis: Good products will ace these; bad products will be obvious – don't have to measure beyond this handful

Method: Identify key security areas (e.g., how do you store passwords, do you provide encrypted connections)

Results thus far: Hypothesis validated, but not enough data to relate fraction of sins committed into # of vulnerabilities

Not applicable where the application gets to rely on an infrastructure (e.g., web server) for security features

Some metrics are retrospective

If acquiring a product, want to know how many security vulnerabilities there are today

Retrospective measures are only valuable as a reputational indicator for early adopters

- Vendor A products have many vulnerabilities
- Vendor B products are rock solid

But if vendor A acquires vendor B, will A’s products get B’s reputation? Or vice versa? What if B acquires A?

Retrospective metrics don’t help with new products or where vendor is unknown

What do metrics measure?

Metrics are of limited value on their own…

$$f(\{\text{(In)security}\} \times \{\text{Popularity}\} \times \{\text{Ubiquity}\})$$

Absolute # of security vulnerabilities

Is the product/vendor (dis)liked by hackers?

Is the product well known/available?

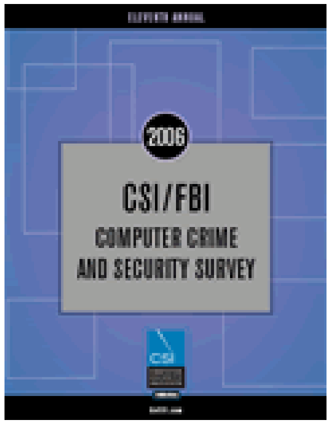
Trying to measure (in)security, or the product of the factors?
# of Bugtraq entries is a measure of the product
# of bugs found in a source code scan is a measure of (in)security

All metrics are created equal, but some metrics are more objective than others

But not all metrics are good metrics…

CSI/FBI study

- Self-selected participants
- No validation of claims, especially for $ amounts
- Claims far more precision (typically 6 digits) than justified by number of responses (typically a few hundred), even if they were randomly selected



Lesson learned

- "Good enough" metrics doesn't mean *any* metrics regardless of quality

What we should do

Open site for public release of data, with product type

- Number of (unfiltered) static or dynamic analysis hits
- Number of Bugtraq or CVE entries / time
- Average education/experience per developer
- # of LOC/developer/time
- % of code that's reused from other products/projects
- % of code that's third party (e.g., libraries)
- Leading security indicators adherence

After data has been gathered for a while, maybe we can draw some conclusions…

In conclusion…

METRICS

JUST DO IT!

Contact information

Jeremy Epstein

Senior Director, Product Security

jepstein@webMethods.com

703-460-5852 (O)

703-989-8907 (M)