Essential Requirements for Data Security in the Context of Software Metrics

Dr.B.R.Sastry[1] **M.V.Vijaya Saradhi**[2]

[1] Director, ASTRA, Bandlaguda, Hyderabad, India.
[2] Assoc.Prof & HOD, Department of computer science and Engineering, ASTRA, Hyderabad, India.

Abstract

By first raising and then dispelling seven common rules about metrics, this paper discusses the requirements and design constraints for a practical system to measure, report and improve data security. Data security will become business-as-usual after the implementation program is completed, but the need for measurement and continuous improvement will persist indefinitely. In other words, we needed more than conventional program or project management metrics. The need for data security metrics was much more pragmatic. Furthermore, intended to embed data security deeper into the academic/corporate culture, meaning that security awareness is an important component. We propose seven rules for data security in the context of Software metrics.

Key words: Data Security, metrics

1. Introduction

Data security will become business-as-usual after the implementation program is completed, but the need for measurement and continuous improvement will persist indefinitely. In other words, we needed more than conventional program or project management metrics.

The fundamental measurement problem Data security, like risk, is a notoriously difficult area to measure, the main problem being how to measure the 'lack of incidents'. The issue is this:

i) If our data security risk analysis is accurate, and if we implement effective data security controls we should avoid, or at least reduce the number and severity of, security incidents.

ii) If we simply measure the number and severity of incidents, we will have some numbers to play with but what will those numbers actually tell us? If the numbers are lower than before we started the data security program, we could claim success ... but ... what if the number and severity of incidents had fallen anyway? If the numbers are higher than before, does that necessarily mean our controls are ineffective? Or could it mean that the threats and impacts have increased and we have not kept pace?

iii) The real issue is one of conjecture. It is practically impossible to measure objectively what might have happened if we had not improved our data security controls.

2. Which things are we going to measure?

This is clearly an important issue but in practice identifying the right metrics is really tricky. We need to take into account the some rules-of-thumb:

i) We shouldn't implement a measurement process if we don't intend to follow it routinely and systematically - we need repeatable and reliable measures;

ii) We shouldn't capture data that we don't intend to analyze - that is simply an avoidable cost. Nobody likes red tape;

iii) We shouldn't analyze data if we don't intend to make practical use of the results. In other words, we need to figure this out first.

We can achieve a lot without expensive solutions or elaborate processes. The true measure of availability, for instance, is the amount of time that an IT service is fully available to the business, expressed as a proportion of the time the business needs that service. The percentage uptime for key IT services is probably already measured by the IT department, especially if those services are covered by Service Level Agreements or contracts. However uptime calculations commonly ignore "planned downtime", "maintenance" and other stated conditions as if they somehow do not qualify as non-availability, which *may*be true but if they only occur outside the agreed service window but not if the system is supporting 24x7 business processes. Also, don't forget that ten one minute outages can be far more disruptive than one ten minute outage. Don't drill too deep. Leave the logistics of data capture to the individual departments or functions closest to the action - simply ensure they follow documented processes and that the data can be validated.

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

3. How will we measure things?

This raises some supplemental questions: where from the data came and how it will be stored? If the source data is not already captured and available to you, you will need to put in place the processes to gather it. This in turn raises the issue of *who* will capture the data. Are you planning to centralize or distribute the data collection processes? If departments and functions outside your control are reporting, how far can you trust them not to manipulate the figures? Will they meet your deadlines and formatting requirements? How much data gathering and reporting can you automate for example by embedding security reporting within application systems.

4. How will we report?

What does senior management actually want? Build your case and seek senior management buy-in to the concepts before you build the entire metrics system. Discuss the purpose and outputs with your managers and peers. The system will inevitably evolve so why not start with that in mind? Start with sample reports and experiment with the style. Provide alternative formats and let management express their preferences.

If you are designing a reporting system from scratch, you have a choice about your style. It may be possible to report differently from other functions in the organization, using different presentation formats as well as different content. This will make data security stand out as something 'special' but at the risk of being seen as no conformant and maybe difficult.

5. How should we implement our reporting system?

When developing metrics, it's worth using the concept of "try-before-you-buy", in other words test out the feasibility and effectiveness of the measurement processes and the usefulness of your chosen metrics on a limited scale before rolling them out across the entire corporation. If you determine the need for new metrics, why not experiment with them for a while? Studies or trials are useful ways to iron-out any glitches in the processes for collecting and analyzing metrics, and for deciding whether the metrics are truly indicative of measure.

Even after the initial trial period, continuous feedback on the metrics can help to refine the measurement system. It is always worth soliciting feedback from the intended audiences about whether the metrics are both comprehendible and useful. Changes in both the organization and the data security risks it faces mean that some metrics are likely to become outdated over time.

Don't be afraid to take opportunities to improve the measurement and reporting processes - small changes in the ways numbers are collected or reported may make the overall process much more efficient without totally destroying the trends, whilst occasional larger changes are justified if the processes simply do not work.

Are the data sufficiently accurate? Bearing in mind rule 7, we may not need perfect accuracy but we definitely do need the figures to be believable and justifiable. Expect management to challenge the source, capture, analysis and presentation of the data, especially if they are under pressure to comply with data security pressures.

6. Setting targets and KPIs

Be aware that if you measure and report something, you are setting yourself up for someone more senior than you to pick out what they perceive to be the Key Performance Indicators (KPIs) and then impose targets on you.

Rules 5 and 6 are particularly relevant here. Before publishing your chosen metrics even as a proposal, it pays to put some time aside to figure out which ones would truly indicate making progress towards the organization's data security goals [7].

7. Seven Rules about metrics

Rule 1: metrics must be "objective" and "tangible"

There is a subtle but important distinction between measuring subjective factors and measuring subjectively. It is relatively easy to measure "tangible" or objective things (such as number of virus incidents or number of people trained) which normally gives a huge bias towards such metrics in most measurement systems, and a bias against measuring intangible things (such as level of security awareness). In fact, "intangible" or subjective things can be measured objectively but we need to be reasonably smart about it (*e.g.* using interviews, surveys and audits). Given the intangible nature of security awareness, it is definitely worth putting effort into the measurement of subjective factors, rather than relying entirely on easy-to-measure but largely irrelevant objective factors [1, 2].

Rule 2: metrics must have discrete values

It is easier to measure and manage things that fall into discrete (preferable binary!) values, rather than those on continuous or even undefined scales. This leads to

another bias towards discrete measures and against things that vary continuously between (often unclear or undefined) extremes.

Rule 3: we need absolute measurements

For some unfathomable reason, people often assume we need 'absolute measures' - height in meters, weight in pounds, whatever. This is nonsense! If I line up the people in your department against a wall, I can easily tell who is tallest or fattest with no rulers or tape measures in sight! This yet again leads to an unnecessary bias in many measurement systems.

In fact, relative values are often more useful than absolute scales, especially to drive improvement. Consider this for instance: "Tell me, on an [arbitrary] scale from one to ten, how security-aware are the people in your department. OK, I'll be back next month to ask you the same question ...". We need not define the scale formally just so long as the person being asked

(a) Has his own mental model of the processes and

(b) Appreciates the need to improve them. We needn't even worry about minor variations in the scoring scale from month to month, so long as our objective of promoting improvement is met. [We'll consider the meaning of 'improvement' in rule 5].

Rule 4: metrics are costly

Metrics or measurement systems *can* be costly to develop, implement and maintain, but they needn't be.

It pays to reuse existing measures where possible and sensible, and wring out every ounce of meaning from the good measures you have. It is surprising how many security-related metrics are already collected for various purposes in the average corporation.

More potential sources of metrics include:

i) Financial data relating to the organization's expenditure on data security, perhaps expressed as a proportion of total IT spend;

ii) Risk based measures such as the proportion of significant audit findings that relate to data security;

iii) Personnel measures from employee satisfaction surveys and so on (could your HR surveys include more direct security awareness measures?)

iv) Customer feedback measures: how often do

v) Management involvement, measured by the proportion of management time spent discussing data security, risk, control and/or governance issues;

vi) Physical security data *e.g.* the total service outage hours caused by unplanned incidents compared to planned maintenance, and the absolute amount of service outage time caused by issues with the physical facilities and services.

The point of rule 4 is that, with a bit of creative thinking, there is probably a large amount of interesting data available to you at little or no cost [3].

Rule 5: you can't manage what you can't measure and you can't improve what you can't manage

Most of us will have heard this old chestnut many times and some of us may even have repeated it, but I contend that it is a rule. There are circumstances where it is provably wrong and most of the time it is sheer nonsense.

Take horse racing for example. It is straightforward to measure the size and weight of a horse - these are stable physical parameters measurements, and the weight at least is directly manageable and "improvable" to some extent (leaving aside the question for a moment about whether improvement means more or less weight, or involves converging on some ideal value).

The moral of that story is that measuring anything makes it easier to drive improvements *but* measuring the wrong things leads to improving the wrong things. The hard part is not measurement *per se* but is figuring out the suite of parameters that need altering and to measure and work on them all, acknowledging that many of the measures are interdependent. Data security is a complex field with ramifications throughout the organization. It is unrealistic to expect to find a few simple measures.

It is important to think long and hard about what actually needs to be improved before building your measurement system, or at least before casting it in stone.

Applying this idea to the data security arena, first of all are you absolutely clear about the purpose of the measurements.

Rule 6: it is essential to measure process outcomes

Data security is all about risk reduction, and risks are notoriously difficult to measure - ask any insurance salesman or actuary. If our controls are effective, incidents should reduce but would they have reduced anyway? Therefore we need to measure the processes of data security not just their outcome, and track our control successes as well as our failures (*e.g.* number of virus or spam incidents *as a proportion of* total inbound viruses or spam's).

Process inputs (*e.g.* the proportion of employees who have been exposed to awareness activities), process activities (*e.g.* the proportion of people regularly updating their antivirus software; audience satisfaction indices for

awareness/training activities) and process outputs (*e.g.* reduction of virus incidents, better audit reports, lower losses) are all worthwhile sources of metrics. The last category most clearly indicates the intended goal of security improvement but there are many influential factors of which security awareness is but one. They need to understand the input and processing measures too.

Rule 7: we *need* the numbers!

The final rule to dispel is that it is essential to generate lots of data, generally meaning numerous objective measures and multiple readings. This argument presses needlessly for additional accuracy and precision, and can emphasize irrelevant metrics purely because the numbers are available. In most practical situations, metrics with more than one or two significant figures indicate spurious accuracy, designed to make people focus on the numbers not the meaning.

There are some aspects of data security and security awareness that simply cannot be measured accurately without an inordinate amount of effort and hence cost. Take for example 'security culture'. Management could conceivably call in a team of psychologists and consultants to measure the culture through questionnaires and interviews with employees, but management should be able to figure out for them with little more than a moment's quiet reflection whether the culture is becoming more or less security-aware. It might be possible to identify parts of the organization where the security culture is more advanced than others, and to use that data as the basis for internal best-practice transfer and developing useful and meaningful data security awareness metrics.

8. Potential metrics

Here is a small selection of metrics that might be worth monitoring and reporting as part of your security awareness program:

i) IT changes statistics (relative proportions of emergency, high, medium and low risk changes; numbers and trends of rolled-back/reversed-out changes, rejected changes *vs*. successful changes *etc*.).

ii) Security-related IT process maturity metrics such as the "half-life" for applying security patches (the time taken to update at least half the population of vulnerable systems - this measure helps avoid the variable tail caused by the inevitable few systems that remain unpatched because they are not in daily use, are normally out of the office or whatever).

iii) Malware statistics (number of viruses, worms, Trojans or spams detected and stopped, number of incidents *etc*.).

iv) Computer audit statistics such as audit issues or recommendations grouped and analyzed by status (closed, open, new, and overdue) and significance or risk level (high, medium or low).

v) Control Self Assessment and other Risk Management statistics - similar to the audit stream but usually cover more of the organization albeit less objectively.

vi) IT Help Desk statistics with some analysis of the number and types of calls relating to data security (*e.g.* password changes; queries about security risks and controls as a proportion of all queries).

vii) IT incident statistics including the number and gravity of breaches, if not some assessment of their costs to analyze, stop and repair the breaches and any tangible and intangible losses incurred. Case studies on serious incidents such as frauds obviously serve to illustrate control weaknesses and also form an effective security awareness-raising mechanism in themselves.

viii) Firewall statistics such as proportion of outbound packets or sessions that are blocked (*e.g.* attempted access to blacklisted websites; number of potential hacking attacks repelled, categorized into trivial/of some concern/critical).

ix) System and network vulnerability statistics such as the number of known vulnerabilities closed open and new; average speed of patching vulnerabilities (analyzed by vendor or in-house priorities/categories).

x) Response to security awareness activities measured by, say, the number of emails and calls relating to individual awareness initiatives [4, 5].

9. Presenting, reporting and using metrics

Presentation of your chosen metrics is just as important as the data content. Does your organization use 'dashboards' or 'balanced scorecards' or notice boards or briefings or what? Again, it is usually worth experimenting a little before settling on a consistent format. If you will be measuring and reporting frequently, the measurement and reporting process should be relatively simple/easy to use/automated, whereas an annual update to the Board can be more labor-intensive.

The frequency of reports depends on organizational norms, the volume and gravity of data available, and

management requirements. Regular reporting periods may vary from daily or weekly to monthly, quarterly, six-monthly or annual. The latter ones are more likely to identify and discuss trends and strategic issues, and to include status reports on security-relevant development projects, data security initiatives and so forth, in other words they provide the context to make sense of the numbers.

Here are some options for your consideration:

i) An annual, highly-confidential Data Security Report for the CEO, the Board and other senior management (including Internal Audit), also known as the '*I told you so'* report. This report might include commentary on the success or otherwise of specific security investments, and of course is the perfect vehicle to point out, subtly, the results of previous under-investment in security (!). Ideally, it is presented to the Board in person, and discussed openly. A forward-looking section can help to set the scene for planned future investments, and is a good opportunity to point out the ever changing legal and regulatory environment and the corresponding personal liabilities on senior managers.

ii) Quarterly status reports to the most senior body directly responsible for data security, physical security, risk and/or governance. Traffic light status reports are common and KPIs may be required, but the Data Security Manager's commentary.

iii) Monthly reports to the CTO/CIO, listing projects participated in and security incidents, along with their $ value (remember, the financial impacts do not need to be precisely accurate - see rule 7 - they are used to indicate the scale of losses).

Avoid focusing too much on the raw numbers but draw out their meaning to the organization. If possible, relegate the numbers to an appendix. Combine numeric measures with feedback comments and suggestions. Pick a key topic or theme for each report. Highlight the relevant numbers and discuss what they really tell you.

10. Conclusion

Data security is a complex area which makes it difficult but not impossible to identify useful metrics. Having raised and dispelled seven rules about metrics, we described the factors that have to be taken into account and suggested a pragmatic approach to the design and implementation of a system of measuring, reporting and improving data security.

References

[1] Rational Software Staff. "Rational Unified Process. Best Practices for Software Development Teams", Rational Software White Paper. TP026B, Rev 11/01,Rational Software, 2001.

[2] Dekkers, C., "Function Points and Use Cases – Where's the Fit?" IT Metrics Strategies, January 1999, pp. 34-36.

[3] International Function Point Users Group Staff. "Function Point Counting-Practices Manual", International Function Point Users Group, Release 4.1.1, Princeton, NJ, 2001.

[4] Jones, C. "Software Assessments, Benchmarks, and Best Practices", Addison-Wesley, Boston MA, April 2000.

[5] Jones, C. "Software Assessments, Benchmarks, and Best Practices", Addison-Wesley, Boston MA, April 2000.

[6] Florac, W.A., Carleton A.D. "Measuring the Software Process", SEI Series in Software Engineering. Addison-Wesley. Second printing, Canada, November 2001.

[7] Bail, W., and G. Vecellio. "Difficulties in Using Cyclomatic Complexity on Software with Error Handling", The MITRE Corporation, Software Eng. Center, Bedford, MA, March 1998. [http://www.mitre.org/support/swee/html/60_bail/sld001.htm]

Dr. B. R. Sastry is currently working as Director, Astra, Hyderabad, India. He earlier worked for 12 years in Industry that developed indigenous computer systems in India. His areas of research includes Computer Architecture, Network Security, Software Engineering, Data Mining and Natural Language Processing, He is currently concentrating on improving academic standards and imparting quality engineering. education. He is widely traveled and a connoisseur of fine arts.

M.V.Vijaya Saradhi is Currently Associated Professor in the Department of Computer Science and Engineering (CSE) at Aurora's Scientific, Technological and Research Academy, (ASTRA), Bandlaguda, Hyderabad, India, where he teaches Several Courses in the area of Computer Science. He is Currently Pursuing the PhD degree in Computer Science at Osmania University, Faculty of Engineering, Hyderabad, India. His main research interests are Software Metrics, Distributed Systems, Object-Oriented Modeling (UML), Object-Oriented Software Engineering, Data Mining, Design Patterns, Object- Oriented Design Measurements and Empirical Software Engineering. He is a life member of various professional bodies like MIETE, MCSI, MIE, MISTE.