GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Implementing a Network Security Metrics Program

Paul W Lowans
GIAC Administrivia Version Number: 2.0

Abstract

If you enter the phrase, "Network Security Metrics" into your search engine, you will probably get less then one page of links. I only had 9 links on Google.com and only two of them were valid. For its importance, to us as network security experts, there is little written on the subject. You need to measure something before you can manage it. Metrics are the only way you can measure the quality of your network and its security. It is the only way you can tell if the improvements to your security are working. You need to be able to report this quality to your management and they mainly understand numbers, percentages, graphs and charts. They need to know the threats to their network and the amount risk in not taking action to correct them. Metrics can help you quantify this information.

Now, enter the phrase, "Software Metrics" into your search engine and you will probably get several thousand links. I got over 22 thousand on Google.com. This is because the software development community has been taking metrics for years on software. They track the number of non-commented lines of code, defects per thousand lines of code, defects per test case and effort, to name a few. Software developers and especially their managers use the metrics to help them determine the quality of the software and it's readiness to release to the customer. This paper will provide you with information on how to implement a security metrics program that is based in part on the already extensive amount of information on software metric programs. You must resist the notion that software metrics is completely different from security metrics. My background is in software metrics and as I researched for this paper, I found the exact same themes being presented about security metrics as I learned in software metrics. In fact, software metrics is not some revolutionary concept, its roots goes back to tracking defects in manufacturing hardware components. It doesn't matter if you are tracking defects in widgets or lines of code or network intrusions. They are all items that need to be tracked in order to understand their nature and how to control them.

What Are Metrics And Their Attributes?

Metrics is a group of measurements that produce a quantitative picture of something over a period of time. Metrics are specific; measurable; attainable; repeatable and time-dependent. An important difference between metrics and measurements is that metrics utilize a baseline as a means of interpreting the results of the measurements.

The quality of our nation's economy can be defined in terms of metrics. The most obvious metrics is the *stock market* but it alone does not provide a complete picture. *Unemployment*, *Gross Domestic Product* and *Housing Starts* are some of the *Key Economic Indicators* that together provide a more complete picture of the state of our economy.

Just as with our economy or software development metrics, there is no one metric that describes the quality of your network security, no silver bullet. You will need to collect many different metrics and compare them to their respective baseline in order to qualify your security improvements.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

In the network security cycle of Prevent – Detect – React – Restore, metrics are tracked in the Prevent and Detect phases. In the React phase, new metrics can be developed to drive improvement to the Prevent – Detect phases.

What to track.

Some suggested metrics to start with:

1. Number of Successful Logons – from security audits.
2. Number of Unsuccessful Logons – from security audits.
3. Number of Virus Infections during a given period.
4. Number of incidents reported.
5. Number of security policy violations during a given period.
6. Number of policy exceptions during a given period.
7. Percentage of expired passwords.
8. Number of guessed passwords – use a password cracker to test passwords.
9. Number of incidents.
10. Cost of monitoring during a given period – use your time tracking system if you have one.

As your metrics program becomes more mature, you will probably want to expand the above list. It may make sense to divide your metrics into two major categories:

Process Metrics —a metric that represents the maturity of a security process. They are best for reporting to management about the quality of your security and improvements. From the above list, #6 is the only example of a process metric. Examples of other process metrics include:

- Percentage of passwords meeting policy.
- Percentage of exposed systems with IDS.
- Number of firewalls per exposed system.
- Number of external users.

Security Metrics —a metric that indicates the extent a security attribute is present. They are best for reporting the state of security to the members of your organization, the collector metric and process implementers. The rest of the above list is examples of security metrics. You can also include the following:

- Frequency of audit reviews.
- Number of compliance with virus updates.
- Number of virus infected components.

Some attributes of metrics are:

- *Understandable* – use the KISS principle, "keep it simple, stupid". If you're the only one to understand the metric then how do you expect your managers to understand the relevance and finance it.
- *Field-tested* – each metric should be confirmed in practice. It may sound like a good metric to take, but, if you can't confirm its usefulness, it's wasted effort.
- *Economical* – automate, automate, automate. If it has to be taken by hand then it costs too much. Try to capture the metric as a natural by-product of the work itself. Security

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

logs can be saved as delimited text files and imported into a database for analysis. Security tools should provide metrics as one of their features. Some of them require a separate application to analyze the logs. This should be purchased along with the tool.

- *High Leverage* – time and money talk best to your manager both in savings and expenditure. The metrics you produce for them should be in this format. Also, metrics should help to pin-point minor changes that provide significant improvements. One way of determining high leverage metrics is to use the *Vital Few* or *80/20 rule*; 80% of improvement comes from eliminating the top 20% of your problems.
- *Timely* – what happened last week, last month or last year is good for baselines but not to plug holes in the security system before a hacker finds it.

Many metrics can be normalized so that systemic changes are factored out. For instance, the number of successful and failed logons will vary if the number of people on your network changes greatly. This is especially true given today's employment market. If you divide the number of network accounts into the logon figures then you get a metric that is independent of fluctuations in the number of network accounts.

Baselines

The main difference between metrics and measurements is that metrics are measurements that are compared to a baseline. You must establish the normal operating level for each of your metrics in order to tell when something is abnormal. In the beginning of this paper, I pointed out that *Key Economic Indicators* are metrics for our economy. Where's the baseline? Whenever an economic indicator is reported, it is always reported as being up or down from *last month* or *last quarter* or *this time last year*. This is the baseline. Previous years are baselines for the present metrics of the economy. We compare the present indicators to previous years, thus determining the present state of the economy. Each yearly baseline is compared to other years in order to establish a trend for the economy.

You must establish a trend for your network security by creating a new baseline whenever you have systemic changes. If your company hires 200 more people or you implement a new security procedure or tool, the normal operation of your system will change and thus your baseline will change. At first this change is compared to the present baseline but as a new trend emerges, it then becomes your new baseline. If your new procedure or tool includes new metrics, then this is included as part of the new baseline.

Objectives of Network Security Metrics

As you design, implement and adjust your metrics program it is important to keep some objectives in mind:

- Collect objective information about the state of the Network's Security.
- Track your organization's progress toward its improvement goals.
- Assess the impact of process/tool changes.

It is easy to loose sight of these objectives and collect data just for the sake of collecting it. Emphasize the business and security results you are trying to achieve. They should be used to

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

drive security improvements and make sure the data you collect get used for constructive purposes.

Don't Fall Into Common Pitfalls.

Drawing upon the experience of the software development community, don't fall into some of the same traps as they have. Metrics should be as painless as possible and definitely shouldn't be shoved down someone's throat. Metrics expert Howard Rubin argues that 80 percent of software metrics programs fail within the first 18 months. Here are some of the common ways they fail.

Lack of Management Commitment

Gain your management's approval from the start. Show them how a metrics program fits into their business goals. They really don't know the *quality* of their security unless they have a way to measure it. Threats need to be quantitative so that management can understand the risk and then they can be accepted, mitigated or transferred.

Measuring too Much, Too Soon

Don't try to measure everything all at once. A deluge of information will confuse you and your management. Start with a complementary set of metrics that you can easily obtain with little effort. Utilize security logs to track logons and use a password cracker to test password strength or number of virus infections and number of virus updates fall into this category.

Measuring Too Little, Too Late

Only collecting one or two items is the other extreme from collecting too much and is just as bad. You won't be able to do much with the data and your management will probably terminate the program thinking it is a waist of time.

Measuring the Wrong Things

Start by collecting metrics that will help you decide where you need to improve your security. Use the goal/question/metric (GQM) paradigm. First, determine your goal, and then list a series of questions that you will have to answer to determine if you are meeting the goal and finally select the metrics to provide you with those answers.

Imprecise Metrics Definitions

A complete and consistent definition of each metric is essential especially if you are not the only one collecting them. If you are in a global enterprise organization, and all network departments are going to compare/combine their data, they must have the same definitions and take data the same way.

Using Metrics Data to Evaluate Individuals

The quickest way to kill a metrics program is to use the data to evaluate people. People are very defensive of their work. Weak passwords or failed logons could be the result of any number of reasons. A one-on-one approach works best to help an individual improve their problem. Only when this doesn't work should you approach their manager. Of course some violations must be reported to management first and this should be called out in your security policy/procedure.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Using Metrics to Motivate, Rather Than To Understand

Metrics are to be used to indicate the quality of your network security, not to motivate people or departments as in a competition. Public signs that say "We have not been hacked in 2 months" should be discouraged. Remember, your global enterprise networks are all linked together, and you are only as strong as your weakest link.

Collect Data that is Not Used

The members of your organization may diligently collect the data and report it, yet they never see it being used for anything. This has happened time and time again in the software development metric programs. People grumble that they spend all this time collecting data, but don't have any idea what it is used for. Include in your metrics plan, reports that are specifically targeted to the members of your organization. They will be interested in seeing the detailed data while your management wants it all distilled into concise information and graphs. It is good practice to solicit the members of your organization for analysis, opinions and suggestions on how to improve the security system. It makes your organization look better if you already have a suggested solution to a problem as you present it.

Lack of Communication and Training

An indication that you have fallen into this trap is, if the participants in the metrics program don't understand what is expected of them. This is also true if there is a lot of opposition to the program. Fear of measurement is a classic sign that the objectives and intent of the program need to be better communicated. If you have not adequately explained the reasons for the measurements or have not trained the collection people in how to perform them, they won't collect reliable data. Create a training class where you can provide this information and field any questions/concerns.

Misinterpreting Metrics Data

View complementary metrics data collectively. Let's take, for example, our logon metric. It may indicate an attempted intrusion into your network but it could also just be someone forgot their new password. Correlating the logons for an individual to when the password changed last and working one-on-one with the individual should provide the correct interpretation of the data.

Creating a plan

There are different approaches to implementing a metrics program. Before you create your plan, you must choose which approach to follow. The approach should depend on what best fits your organization and not which is fastest and cheapest. If your organization has a software development community and they have experience with *Software Engineering Institute's* (SEI) *Compatibility Maturity Model* (CMM), then it may make sense to follow the *International System Security Engineering Association* (ISSEA) *Systems Security Engineering Compatibility Maturity Model* (SSE-CMM). You can draw on their experiences to guide you. The approaches below are by no means the only approaches to choose from. Nor is any one approach better then another. They are presented only to expose you to some of the many choices you have.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

SSE-CMM

In the software development field, one of the most prominent quality programs is the *Software Engineering Institute's* (SEI) *Compatibility Maturity Model* (CMM). The software CMM helps an origination develop cost effective consistent quality software.

There are five levels to the CMM. Level one is ad hoc, the organization has a few procedures in place and quality software depends on heroes or developers that are just plain good at writing software. The top level is five where there are feedback processes and procedures in place that provide for continuous improvement. The software CMM has improved the quality of the software and is not a metrics program, but as an organization matures up through the levels, more and more metrics are needed to measure the quality of the code. The software CMM is so well accepted that many government agencies require a certain CMM level before awarding contracts.

SEI has not created a CMM for security, but the *International System Security Engineering Association* (ISSEA) has created the *Systems Security Engineering Compatibility Maturity Model* (SSE-CMM). Like the software CMM, the SSE-CMM consists of five levels; each builds upon the previous level until the fifth level is a system that is continuously improving. Also like the software CMM, metrics plays an important roll in measuring the quality of the security procedures and processes. The objective of the SSE-CMM is to advance security engineering as a defined, mature and measurable discipline. It is important to keep in mind that the SSE-CMM like the software CMM does not require you to follow a prescribed methodology or process. It does require that you document your processes and that those processes are institutionalized in the organization.

Level 1"Performed Informally"

Focuses on the security process an organization has in place. That *Base Practices* are performed.

Level 2"Planned and Tracked"

Focuses on project-level definition, planning and performance. Metrics are defined, planned and taken to measure performance and establish baselines.

Level 3"Well-defined"

Focuses on disciplined tailoring from defined processes at the organizational level. Metrics are expanded to the global enterprise level.

Level 4 "Quantitatively Controlled"

Focuses on measurements being tied to the business goals of the organization.

Level 5 "Continuously Improving"

Leverages all the management practice improvements from earlier levels and emphasizes cultural changes that are needed to sustain the improvements. Metrics are used to drive security improvement decisions for the global enterprise.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Six Sigma

Originally developed at Motorola to improve the quality of manufacturing, *Six Sigma* is a rigorous, focused and highly effective implementation of quality principals and techniques. The *Six Sigma* standard is 3.4 problems per million opportunities. The model that drives *Six Sigma* is known as *Define-Measure-Analyze-Improve-Control* (DMAIC).

- **Define** – the goals of the improvement activity
- **Measure** – the existing system. This establishes a baseline.
- **Analyze** – identify the gap between present performance and desired goals.
- **Improve** – the system by eliminating or minimizing the gaps.
- **Control** – the new system. Institutionalize the improved system.

Six Sigma utilizes a hierarchy of fulltime agents that control the process. It utilizes a top-down implementation that includes a *Master Black Belt* – to provide the technical leadership; *Black Belt* – who are actively involved in the organizational change and development and *Green Belt* – who are the project leaders capable of forming and facilitating Six Sigma teams. The effort has to be lead by the CIO or department head (Vice-President) who is responsible for global enterprise network.

Defect Elimination Approach

A project or process manager creates a *security profile* of the network environment that includes a set of measurable data or *defects*. Processes are put into place to quantify the defects. These processes include procedures to minimize or eliminate it. As the defect decreases the system becomes more secure. This model has been extensively used in manufacturing to control quality but has been adopted for use in other non-manufacturing areas.

Components of a Plan.

Each component is important for the success of a metric program. The policy is where management gives support and the security document spells out what metrics to take, how to take them and who will take them.

Security Policy

A section on Security Metrics must be included in the "Security Policy" if you are serious about taking security metrics. The section should include providing for the resources, and training to take the metrics data and define who's responsible/accountable for the metrics program. This is where management agrees to support the metrics program.

Document the Network Security Process

Your network security process must be documented to explain just how your organization provides network security in accordance with the policy. Since the policy now includes providing for a metrics program, the Network Security Process explains what the metric plan is and how it will be implemented.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

State the Goals

The goals tell why you are taking the metrics, most important of which is to improve the quality of your security system.

"Reducing the response time for detecting an intrusion, improve the password protection of your accounts, and reduce risks?"

Define Metrics Required to Reach Goals

Tells what you're going to measure in order to determine when you've reached a goal.

If you want to improve the password protection of your accounts then you need to know what metrics to collect to determine what a "good" password is.

"How long the password is, is it made up of alphanumeric and how long does it take a password cracker to crack it."

Identify Data to Collect

Now that the metrics are defined, they can be expanded into the raw data to collect.

"For passwords, you need to collect for each account name, the number and type of character makeup the password is and measure how long it takes to crack."

Define Data Collection Procedures

Tells you how and when you are going to collect the metric, including what tool to use. Also, who is responsible for collecting the data, where it is to be stored and how to verify it. If a tool is used, then you need to include the tool's location, version and setup in the definition. Scripts should be treated the same way.

"On a weekly basis, each network administrator will run LC2 password auditing software on their respective PDCs. They will record the account name, the number and type of characters, and how long it took the tool to crack the each password. This information is to be stored in the SecurityMetrics.mdb database."

Define Training

One of the most important aspects of collecting metrics is consistency and training is the best way to accomplish this. This is especially true in a global enterprise so that the metrics collected by all the individual groups can be correlated.

"Those using LC2 will take a half day course provided by the Security Officer."

Define the Feedback Mechanism

Another very important part of a good metrics program is feedback. The process should define where, how often and to whom the feedback is to be given with provisions for emergency reports. This feedback needs to be at all levels of the organization, starting at the bottom and going up to management. Recommendations for improvements are received and considered. When you start at the bottom, you can bring the improvements with you when you present the data to management. It looks a lot better for you and your organization when there is a problem, if you also have the solution.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assemble a Metrics Toolset

All tools and scripts should be conveniently stored on the network so that anyone who needs to can access them. Access should be "read only" so they cannot be changed unless specifically authorized. This is especially true of scripts. You might even consider a formal *Software Configuration Management* system for scripts to track their changes.

Create a Metrics Database

The database should be easy to use, flexible, interface with or include a graphical reporting routine to enable professional graphs and charts. It should be capable of storing large amounts a data for historical purposes.

Tips On Presenting Your Results.

There are three kinds of lies: lies; damn lies; and statistics. The same can be said about metrics. Don't manipulate the data to suit your needs, it can only hurt you and the metrics program in the long run.

Graphs and data sheets should include the baseline for comparison.

Keep your reports to management simple; don't deluge them with the raw data. Distill the information into graphs or datasheets in the form of high level data or percentages. If you are presenting the information at a meeting, give handouts to the audience and have backup data for yourself in case they ask questions.

One Final Metric

In the software metric world there is a proven metric, it costs ten times as much to correct a defect the customer sees then if it was caught in code inspection. Track the amount it costs to correct security vulnerabilities. Then when you get hacked, track how much is lost and how much to recover. You'll know what to do with these figures the next time management asks why security costs so much.

Conclusion

Metrics need to be a required part of Network Security. All process improvement plans must have feedback loops in order to drive quality as high as it will go. Metrics are the key enabler to improving your network's security. It is the only way you can determine the quality of your security.

As it is said, *improvising is the mother of invention* and since there isn't much written on Network Security Metrics, then you need to draw upon other metric initiatives such as those used by the software development community. Metrics programs are worth the effort and investment especially compared to the losses.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

References

Lemberg, Paul. "The Vital Few" Lemberg + Company
http://www.lemberg.com/80-20.htm

Jelen, George. "SSE-CMM Security Metrics." NIST and CSSPAB Workshop, Washington, D.C., 13-14 June 2000.
http://csrc.nist.gov/csspab/june13-15/jelen.pdf

Wiegers, Karl E. "Software Metrics: Ten Traps to Avoid" 5 Oct. 2000
http://www.processimpact.com/articles/mtraps.pdf

Pyzdek, Thomas "The Six Sigma Revolution"
http://www.qa-inc.com/knowledgecente/articles/PYZDEKSixSigRev.htm

Craft, James P. "Metrics and the USAID Model Information Systems Security Program" 14 June 2000
http://csrc.nist.gov/csspab/june13-15/Craft.pdf

Lindquist, John; O'Shea, Connie; Ferraiolo, Karen and Jelen, George. "SSE-CMM: Model & Appraisal Method Summary" April 1999
http://www.issea.org/docs/ISSEAPamphlet.pdf

"Systems Security Engineering Capability Maturity Model Version 2.0" April 1, 1999
http://www.sse-cmm.org/Papers/SSECMMv2Final.pdf

Yourdon, Ed. "Software Metrics: The Next Productivity Frontier" Computerworld Australia September 1993

"SPC Resources – 8 Step Metrics Program"
http://www.spc.ca/resources/metrics/8steps.htm

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.