Security Metrics
J. L. Bayuk

It may have been the Melissa. It may have been the Microsoft break-in. It may have been a penetration test or an internal audit report. Whatever the reason, management is now ready and willing to fund information security efforts. The question is no longer, "Should we spend?" Risks have been identified, budgets have been forecast. The question is now, "What should we spend it on?"

Information Security has joined the ranks of Configuration Management and Quality Assurance. It is one of many high priority tasks competing for Information Technology (IT) risk management dollars. Configuration management systems reduce the risk of production outages. Quality assurance systems reduce the risk of customer dissatisfaction.   Information security systems reduce the risk of information damage and disclosure. These efforts are similar in that an investment can reduce risk.   But they differ in the methods available to determine whether the investment has paid off.

For example, if money is spent on a new help desk system, IT managers can usually come up with a set of statistics that show that customers spend less time on hold, their questions are more quickly answered, and they are more satisfied with the company's service. What measurable benefits can information security investment provide?   What is the basis for a fact-based, genuine, indisputable proof that security is working? The test of whether a security metric is successful is the extent to which it can provide answers to these questions. Security metrics should be able to help IT management justify IT security efforts.

A recent survey of approaches to security metrics revealed as wide a variety of approaches as there are interpretations of the word "secure".[1] This article explains the approaches that have been tried, compares their benefits in answering the above questions, and recommends an overall strategy to information security metrics.

Security Measurement Models

Despite the lack of consensus over what makes a system "secure," many professionals are asked to make that determination on a routine basis. A practitioner will adopt a "Security Measurement Model" designed to provide an assessment of the overall security of the IT Environment. The model adopted will enable the practitioner to provide an "Industry Standard" security assessment.

Industry standard security assessment models fall into one of five categories:

- External Audit
- Internal Audit
- Capability Maturity
- Risk Analysis
- Defect Elimination

External Audit

The external audit model assumes that there are "best practices" available on how to secure a given environment. "Best practices" are loosely represented by the recommendations of publications such as this one, with some respect to the extent those recommendations are successfully implemented in similar IT environments. The external audit holds management accountable for implementing controls that

[1] Security Metrics Workshop, Computer System Security and Privacy Advisory Board, National Institute of Standards and Technology, Gaithersburg, MD, June 13-15, 2000. Proceedings at http://csrc.nist.gov/csspab/june13-15/sec-metrics.html

protect information systems assets using these best practices. Industry standard control objectives provide the standard unit of measure.[2]

A control objective is a specific, measurable goal that IT management is expected to have set to reduce risk. By definition, a control objective is measurable. An external audit is the process of measuring the extent to which control objectives are met. The measurement is done by gathering evidence of best practices established by management that contributes to control objectives. Specific audit steps determine exactly what an auditor will do to gather evidence of best practices.

The external audit model measures security by comparing the level of management's control over the current systems environment to that which would result if best practices were followed.[3] The final measurement in an external audit is a list of security weaknesses, or defects, that must be corrected in order to bring systems to an acceptable level of risk.

Financial auditors developed this model long before computers existed.[4] External auditors need to ensure that they are holding their auditees to standards that will provide integrity to financial statement processing. They are also liable for attesting that assets are effectively safeguarded. The best practice comparison provides comfort that they are measuring security with as thorough as possible due diligence.

[2] See Control Objectives for Information and Related Technology Framework (CoBIT), issued by the Information Systems Audit and Control Association (ISACA), 1998.

[3] Because one obvious best practice is managing security to keep out unwanted intruders, penetration studies fall under this model.

Internal Audit

The internal audit model assumes that management has adopted a set of control objectives designed to secure information systems assets. The difference between it and external audit is that the control objectives measured are expected to originate from the organization being audited. The objectives are expected to be the result of a formal risk analysis process. It is assumed that management has assessed the risk to information systems, designed strategies for dealing with it, and followed those strategies. If this assumption is met, an internal audit will measure the extent to which management has succeeded in these efforts.

Where it cannot find evidence of management setting control objectives, an internal audit will revert to industry standard control objectives and the external auditor's comparison of current systems control environment to that which would result if "best practices" were followed. Like the external audit model, the internal audit model will gather evidence of activity established by management that contributes to control objectives. It will similarly provide metrics in the form of a list of security weaknesses. The internal audit model may in addition provide percentage measurements showing the extent to which management controls were achieved in comparison to prior year audits and to similar IT environments within the company.[5]

Capability Maturity

The capability maturity model assumes that organizations committed to securing their environment will formally adopt a process for so doing. It further assumes that they will document the process, do what they document, verify that it is done,

[4] Bayuk, J.L., *Stepping Through the IS Audit*, Information Systems Audit and Control Association (www.isaca.org), 2000, Chapter 1.
[5] Bayuk, J.L., "Information Security Metrics - An Audit-Based Approach," Security Metrics Workshop, Computer System Security and Privacy Advisory

and be able to measure process improvement. The foremost champion of this approach has been the International Security Systems Engineering Association[6], but several corporations and government agencies have adopted similar approaches.

All have stages of development, usually five stages. The lowest stage is "informal" or "immature." In the middle is "quantifiable," the stage at which security can be measured. The highest stage represents continuous improvement. All proponents of this approach admit that the hard part is to define a process that everyone will agree will result in a secure environment. Few if any have been defined past the "quantifiable" stage.   All incorporate some aspects of the audit models in their measurements, an audit is often used to determine which stage of development an organization has achieved.

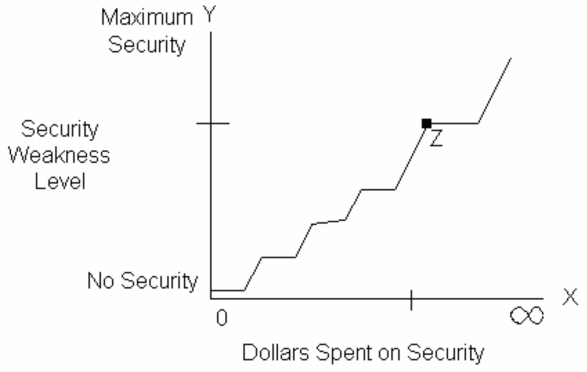
Metrics provided by the model are ratings that indicate the extent to which the organization has reached a certain stage of development. This makes the capability maturity model of security assessment attractive to internal standard-setting organizations, information security consultants and security product vendors. The attraction lies in the fact that a capability maturity model approach to measuring security could provide consultants and vendors with comfort that their advice and products will be used in a context where they will be continually refined and thus most effective. It also could provide clear standards by which consulting advice and security product usefulness may be easily judged. Success could be measured by whether the consulting advice or product usefulness brought the organization to the next stage.

Board, National Institute of Standards and Technology, Gaithersburg, MD, June 13-15, 2000.

Risk Analysis

The risk analysis model assumes that there is a known dollar figure that represents how much it would cost to "completely secure" the IT environment. Picture a two-dimensional graph where the X-axis is the number of dollars spent on information security projects and the Y-axis represents the security of the environment (see Figure 1). The risk analysis model assumes that if no money is spent on IT security, no security will be achieved. It also assumes that there is a measurable continuum from that point up to unlimited money and maximum security. Ticks on the Y-axis are associated with security weaknesses that threaten information systems assets. Points in the resulting curve associate X and Y. X is the number of dollars required to bring the security of the environment to a level that addresses the security weaknesses measured by Y.

Figure 1: Risk Analysis Model



Call a given point on the graph "Z." The risk analysis model presumes that it is possible to calculate the value of the assets that would be put at risk due to the level of security weaknesses measured by the Y-axis at point Z. Call the result of the asset value calculation "AV." AV should be compared to the value of the X-axis at point Z. The comparison will tell if it is a better idea to spend "X" on

[6] Bartol, Gallagher, and Givans, *Developing and Applying System Security Engineering Metrics*, Systems Security Engineering Conference, February, 1999

security than to live with the risk of losing "AV." The idea is to figure out how much money needs to be spent to achieve a "reasonable" degree of security.

The risk analysis model is often used where the dollar amount of the risk is obviously quantifiable, such as a loss of revenue stream from a given order-processing system or manufacturing line. The dollar amount at risk is compared to the usually trivially insignificant price of a security improvement.

This type of measurement approach not attempt to portray detail on how to measure the level of security weakness to the overall environment. However, it is useful in presenting cost versus risk analysis required for IT-related insurance policies. It allows risk management specialists to view systems not as individual points of potential vulnerability, but as statistical aggregates. A group of systems sitting in an IT data center supported by a reputable company that pays industry standard wages to employees can be expected to have vulnerabilities resulting in an average dollar amount of loss equal to reported average losses of other sets of similarly situated systems. Risk management specialists need information security measurements only in order to set dollar amounts on security-specific service levels agreements.

Defect Elimination

The defect elimination model assumes that there are specific measurable parameters inherent in an IT environment that reflect its "security profile." The security profile itself is then defined in terms of the set of all measurable parameters.[7] The defect elimination model assumes that a mechanism exists to measure these parameters, that a numeric is associated with the measurement, and

(http://www.sse-cmm.org/librarie.htm).

[7] This approach is thus circular but complete. It of course must be combined with a continuous improvement process in order to remain effective (e.g., Six Sigma).

that there is a function that, when applied to the numeric, approaches zero as the IT environment becomes more secure.

Defect elimination proponents tend to be project and process management specialists. Defect measurement and subsequent elimination is a time honored manufacturing quality control technique that has widely recognized potential for application to other-than-manufacturing processes.[8] Project managers that have used the technique successfully in the past for a variety of projects apply it to IT security metrics. Defects measured include everything from the number of system administrators with no formal security training to the number of alarms generated by intrusion detection software.

Model Comparison

Note that the lead sentence in each of the preceding sections starts with the phrase, "The X model assumes that ...." The phrase is meant to call direct attention to the fact that models for security measurement rely on assumptions. The assumptions are a reflection of the standards by which the measuring organization is comfortable labeling an IT environment "secure." The assumptions for each model are, respectively:

- External Audit - there are "best practices" available on how to secure a given environment.
- Internal Audit - management has adopted a set of control objectives designed to secure information systems assets.
- Capability Maturity - organizations committed to securing their environment will formally adopt a process for so doing.
- Risk Analysis - there is a known dollar figure that represents how much it would cost to "completely secure" the IT environment.
- Defect Elimination - there are specific measurable parameters inherent in an IT environment that reflect its "security profile."

Note that all models measure security by comparing the environment to an assumed pre-existing standard.

[8] Harry and Schroeder, *Six Sigma, The Breakthrough Management Strategy Revolutionizing The World's Top Corporations,* Doubleday, 1999.

Investigative Approaches

Assumption-based measurements are not necessarily bad, as they help serve to clarify objectives. And it is evident that models for information security measurement do have specific objectives. The purpose of the model is to provide a way to measure the extent to which the objectives are met. There is usually a role for an "investigator" to do some research and some digging into the IT environment in order to come up with the measurement corresponding to these objectives. Hence, one approach to information security metrics may be called an "investigative" approach.

Using the investigative approach, some quantity of elements is manually gathered. Each is reduced to a numeric or boolean value. A value may be multiplied by some weight. There is a pre-determined "best score" to be achieved. The measurements are compared to that score. Metrics are thus compiled by a person as he or she searches for specific evidence.

All models of Information Security measurement may be accomplished using the investigative approach. Audit and capability models require auditors to make up audit tests, list those that have failed, and decide on whether vulnerabilities make the final report based on some weighting criteria. Risk analysts gather parameters related to the business use of systems and reduce them to dollar amounts. Defect elimination specialists identify situations that indicate defects, and develop ways to quantify them. These approaches follow the respective model, and also require a person to gather and analyze evidence in order to produce security metrics.

Automatic Approaches

The opposite of an investigative approach is an automatic approach. In the automatic approach, some quantity of elements is automatically gathered. The numeric is stored in a database and available for reporting. Reports may or may not use weights to present the final overall security measurements for an IT environment. Metrics are thus the direct result of measuring mechanisms interacting with systems.

The defect elimination model alone allows an automated approach to security metrics. Though the model can be implemented as investigative, if combined with the requirement that parameters that comprise a security profile coincide with parameters that may be automatically measured, it can be solely automatic. Even proactive measurement of configuration parameters can be viewed as measuring the absence of defects. The absence of human intervention in the measuring and reporting process renders this type of defect elimination method the sole example of the automatic approach to security metrics.

As noted above, the defect elimination approach can also include aspects of an investigative approach. Often it is necessary to evaluate the extent to which a given alarm or log entry is really a security incident. If independent judgement is required to make this determination, it falls back to an investigative approach. Yet, if the determination can be proceduralized, then theoretically it could one day be automated and could be counted as an automatic approach.

Success Criteria

The test of whether a security metric model is successful is the extent to which it can provide answers that help IT management justify IT security efforts. IT management must be able to prove that IT security is effective. A successful security measurement method can provide metrics to answer the previously considered questions:

- What measurable benefits can information security investment provide?
- What is the basis for a fact-based, genuine, indisputable proof that security is working?

The successful combination of security measurement model and approach should allow IT management to claim concrete benefits comparable to customers spending less time on hold, their questions more quickly answered.

Table 1 summarizes the information security measurement models described above with respect to the metric produced by the corresponding measurement approach.

Table 1: Metrics produced by Models

Approach: Measurement Model:	Investigative Approach	Automatic Approach
External Audit	Vulnerability Listing	-
Internal Audit	Vulnerability Listing	-
Capability Maturity	Organization Rating	-
Risk Analysis	Recommendation for Spending	-
Defect Elimination	Summary of manually and automatically measured variables.	Summary of automatically measured variables.

The metric produced by an investigative approach will be a vulnerability listing, organizational rating, recommendation for dollars spent, or some combination of manually and automatically generated lists of variables. These will drive efforts for security improvements. A follow up investigation will produce the same output. Comparing the original metric to the follow-up can result in the identification of improvement. This approach does then meet the criteria of measuring security and providing assurance that security measurements are working. But there are three shortcomings with this approach.

The first is that there is much subjective judgement in the mapping of control objectives, capability metrics, or vulnerability levels to actual investigative tests or evaluation procedures. There also may be subjective judgement in evaluating results of investigative tests. Requiring strict standards, guidelines and procedures from all investigation teams may mitigate this subjectivity.

Unfortunately, the requirement for requiring strict standards, guidelines and procedures leads directly to a second shortcoming. The time it takes to fully document all procedure steps becomes a significant expense. Side by side with system engineers designing the next evolution of technology, an IT department would be funding independent efforts to develop security measurement programs, tools, and techniques. Then it would fund teams of investigators to constantly cycle through all systems, measuring security.

Even if the requirement for requiring strict standards, guidelines and procedures, and rotating investigations was not prohibitively expensive, the investigative approach has another inherent shortcoming. The deliverable of investigations are lists or ratings determined through comparison to pre-established assumptions at the time of the investigation. It is always going to be a snapshot at a given time. But security vulnerabilities often appear as a result of normal system maintenance. These need exist for a matter of minutes to be exploited. The exploit could happen in between investigations or when the investigators were looking elsewhere. Hence, though the investigative approach can answer some questions about whether a security expenditure was effective, an investigative approach could never provide complete assurance that the system security measures are working.

As in the investigative approach, the automated approach has a shortcoming in that there is much subjective judgement in establishing the measurement process. In this case, the subjective judgement is in mapping of security objectives to automatic security measuring mechanisms. The mapping must somehow ensure that parameters that indicate whether systems are secure coincide with the parameters that are automatically measured. To mitigate that shortcoming, a demonstration that such automatic measurements are possible is a requirement for implementing the approach.

Once parameters to measure are decided, automated security metrics can be built into all system requirements and engineered by the same IT engineers who are charged with deploying the systems themselves. Hence, the second shortcoming of the investigative approach does not apply to the automatic approach. The automated approach is not only less human-resource intensive at the onset, but substantially less human-resource intensive on an ongoing basis. Also, since the automated approach measures pre-establish security parameters of the IT environment at periodic time intervals that may be set at any interval desired in order to detect security-related events, it does not share the third shortcoming of being too brief a snapshot.

Of course, where metrics are immediately required and the automated measurement capability does not exist, IT management is better off with outside investigators. But if the choice is between developing an internal investigative or an internal automatic approach, the time and effort spent in one or two internal investigations could more efficiently be spent instead developing parameters and automated measuring mechanisms for the automatic approach. In conclusion of the comparison, no approach is ever perfect, but if diligently implemented, the automated approach can bring IT management closer to a fact-based proof that security is working than the investigative approach.

Measuring Security

The conclusion may seem hollow to those who have no automated mapping from parameters that indicate whether systems are secure to the parameters that are automatically measured. Hence, the rest of this article will draw out other consequences of this approach for measurable and consistent approaches to security measurement. It will examine the key factors of the security parameters that provide reliable metrics. By concentrating on choosing parameters that comprise a security profile and are measurable, heretofore obscure security processes become more goal directed and thus more understandable to the rest of the IT community. They become more focused on achieving well-defined objectives of IT management.

Measurement Systems

Recently, a slew of security start-up firms have had the idea to integrate security intrusion detection tools into "security monitoring platforms." The pattern is similar is most cases. A few partnerships with major vendors are created, the system reduces logs and alert messages from each system to a set of standard patterns to be monitored and sued to generate alerts. Unfortunately, the commercial products are limited in that they support a small set of security products. Even if they support plug-ins, those features lag well behind existing non-security-specific systems and network management platforms. The patterns tend to be focused on intrusion detection as opposed to Security Metrics.

Nevertheless, these vendors have a good idea, and a simple product to develop. Security metrics are specialized parameters and need a specialized platform. As described in the sections that follow, correlation among reports from different platforms and security products is necessary to measuring security. Generating correlation reports must be done on some system. The system must meet some basic requirements:

- The measurement system itself must be secure
- The measurement system must be able to access the actual system data, not pre-processed summaries or manually-derived evaluations.
- The measurement system must use that data to generate security metrics.

Assume the actual parameters for measuring security are identified, it suddenly becomes obvious that any data center that has scale enough to have economized on monitoring tools already has a system, and perhaps several systems, for measuring security. These are the same mechanisms that have been used to alert on disk full conditions, process failures, and downed router interfaces. Each monitored component required some systems engineer to devise a configuration to integrate some alerting feature on the monitored system with some message receipt feature of the alerting system. Most alerting features support many monitoring protocols.

The more use that can be made of existing monitoring systems, the easier the integration of security metrics. Even when the final security metrics report comes from a central security server, complexity is reduced and efficiency achieved by relying on the existing infrastructure to deliver security-relevant information from a system from which it is already gathering performance and other resource data.

For example, consider a Security Measurement system that must report on all network systems, all operating systems, and some application systems.   Consider two alternative communication architectures for the system, Scenario A and Scenario B (as in Figure 2). Scenario A requires the security system to maintain mechanisms for individual communication with all three types of systems. Scenario B takes advantage of the fact that the targeted Network and Operating Systems already have communications mechanisms in place with other monitoring systems, and so receives its data by proxy through them.

Figure 2: Security Measuring Scenarios

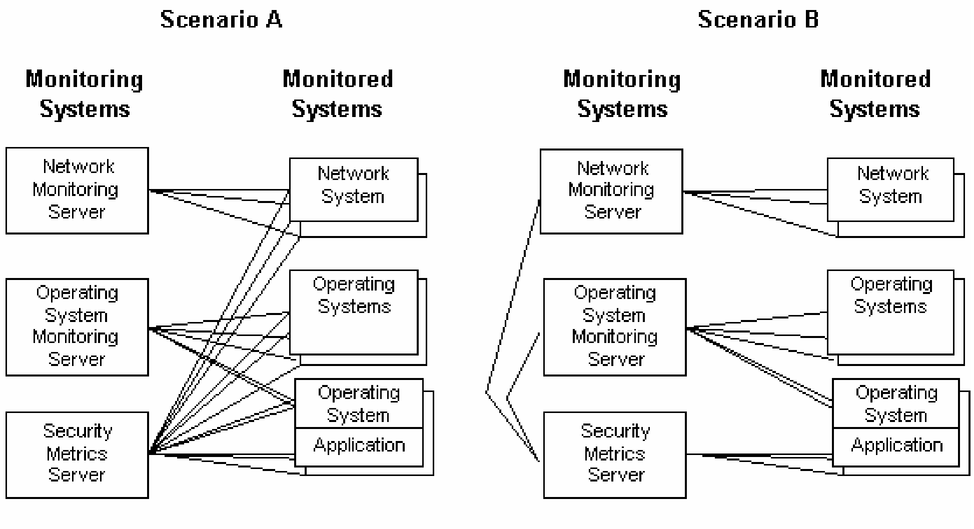


Figure 2 shows that Scenario A requires individual communications links with all monitored systems, while Scenario B trades several individual system links for just two links: one to the Network Monitor and one to the Operating System Monitor. This architecture alleviates requirements for a potentially resource-intensive security-specific messaging infrastructure at the expense of a few extra data fields in the existing monitoring system, and perhaps an extra process running on the monitored systems.

An oft-cited downside to measuring through proxy is that the security server is relying on intermediary systems that may not have the same integrity standards for data preservation as the security measurement system itself. Yet this argument assumes that nothing can be done to bring the monitoring system up to standards where it can be trusted to provide proxy services. Depending on the size and complexity if the IT environment, efforts to increase the reliability of the monitoring servers are almost always preferable to adding message processing and network traffic on production systems.

The remainder of this article will assume that there is some mechanism in place to collect data relevant to security metrics. It will concentrate on the third requirement of a security measurement system. The measurement system must use collected data to generate security metrics

Measurement Techniques

There are three basic types of mechanisms available for measuring security. Each mechanism measures a different aspect of the IT environment's security profile.[9]

Logs:	System and user activity
Configuration:	Variable settings
Service:	Methods of system access

For example, an unauthorized addition of a new account to a system is generally accepted as a security defect. It could be measured through a log monitor, a configuration monitor, or a service monitor.

To measure this type of defect with a log monitor, a process would have to examine all logs of new account additions and compare them with the evidence from the authorization process that was performed by the administrator assigned to add new accounts.

[9] For more detail on these monitoring methods, see Bayuk, J. *Infrastructure Monitoring Challenges*, 22nd Annual National Information Systems Security Conference. Baltimore, MD, October,1999.

To measure this type of defect with a configuration monitor, a process would have to detect changes in the snapshot of user lists on a periodic basis and compare these lists with the evidence from the new user authorization process.

To measure this type of defect with a service monitor, a process would have to try to gain access to the systems using names of users who are not authorized to use the system.

Only the second method will allow the automated measurement of an unauthorized addition of a new account. The log method would miss any unauthorized accounts added through a process that did not create logs. The service method relies on guessing user names so the measurement would never be complete. Each security defect that makes up the security profile of the environment must be similarly mapped to an appropriate security measurement mechanism. Once a mechanism is defined for the metric, detailed requirements may be written to accomplish their implementation.

These requirements may have consequences not only for log, configuration, and service monitors, but also for IT process. For example, if IT process does not currently keep on-line records of account approvals, the second method for correlating those with actual accounts added may not be possible to implement. The process must be modified to make the authorization records automatically available prior to automated measurement of the " unauthorized addition of a new account" metric.

Defining the Metrics

If the assignment to measure security was treated as any other system engineering task, the appropriate first response is a requirements analysis.   Proceeding with the defect elimination model, the purpose of the system is to measure defects in system security. So the requirements analysis must be focused on mechanisms with which to detect security defects. A necessary precursor to the requirements analysis is a definition of security defects.

Security is generally defined as activity directed at preventing or detecting harm to systems, and also at providing recovery mechanisms in the event that protection mechanisms fail. So there are two types of activity that constitute a security defect:

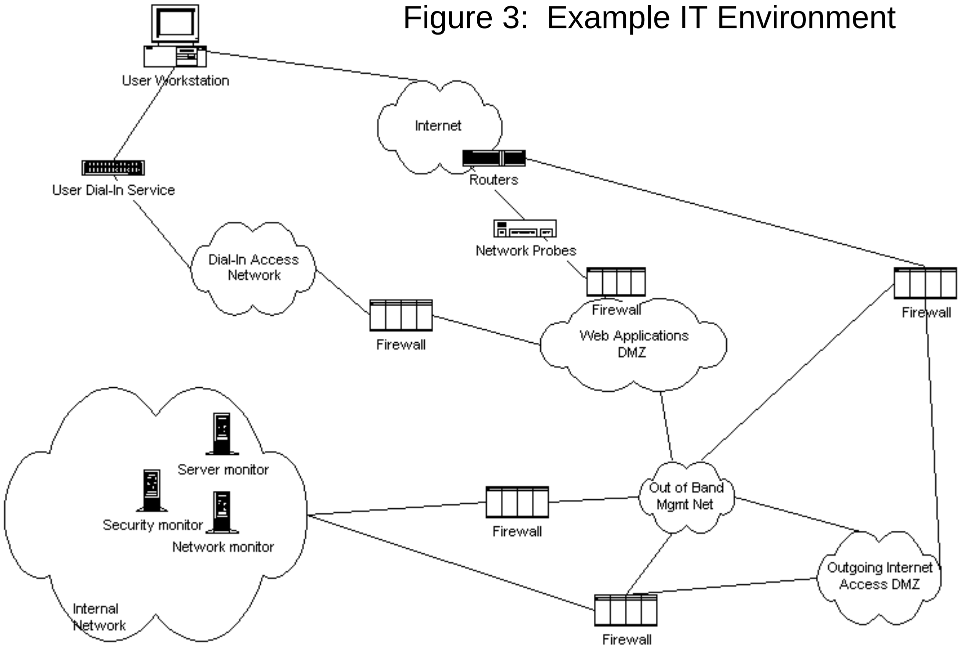
Corruption: The misconfiguration of a mechanism that prevents, detects, or facilitates recovery from harm to systems
Intrusion: The bypass of those mechanisms

To detect the corruption, one must know what mechanisms have been deployed that prevent, detect, or recover. To detect the intrusion, one must know what activity constitutes an intrusion. Of course, it may be that system security corruption happened as the result of an intrusion. But it is often the case that corruption results from authorized access, so a correlation to intrusion cannot be assumed. System corruption detection mechanisms may be independent of intrusion detection mechanisms.

All three categories of monitoring may be necessary to measure both corruption and intrusion with respect to a given IT environment. For example, see the diagram in Figure 3. The example shows an environment where users may access web applications either by dialing in or through the Internet. Security-relevant components of the infrastructure include dial-in devices, routers, Firewalls, web servers, network probes, and existing management networks and servers.[10]

[10] To keep the diagram simple, does not show every piece of network equipment that must be monitored in order to completely secure all network paths. However, the following discussion may be assumed to cover those not specifically identified.

Figure 3: Example IT Environment



Individual components of this architecture have readily available logs, configurations, and services available to be used in defect measurement.

Corruption

It seems easy to list out measurements that indicate corruption of security because all that is required is knowledge of what security mechanisms are deployed. Wherever a given system has features that allow monitoring of system integrity, those same mechanisms can provide an indication of system corruption. They may be logs, configuration, or service monitors. Where platform specific corruption detection tools are not readily available, any systems component can be monitored for corruption via a configuration monitor. All that is required is a snapshot of current values for specific sets of configuration parameters that define the security of each architecture component. The snapshot is compared to previously agreed upon configuration values, and exceptions constitute defects. If the configuration data is kept in a non-readable format, a periodic database query may be used to determine whether the parameters have been modified. If no modification detection tool can be built into the infrastructure component, parameter values can be copied to the Security Metric Server where a generic modification detection tool (e.g. Tripwire), can detect changes off-line.

Some example metrics indicating corruption in the security of the example environment would be:

Logs:

Application Servers: Failed database object requests.

Dial-In Servers: IP address allocation log without corresponding authentication log.

Firewalls: Long lapses in log activity.

Operating Systems: Logs of security processes starting or stopping other than during system reboot.

Web Servers: Web Page not found records, Java errors.

Configuration:

Application Servers: Changes in any file or configuration parameter and no correlation with maintenance activity.

Dial-In Servers: Changes in variables that specify authentication types and allowed network access connectivity.

Firewalls: Rule changes that cannot be correlated with authorized maintenance procedures.

Monitors: Unexpected changes in inventory of systems sending metric data, as well as inventory of data received.

Operating Systems: Changes in configuration outside of normal maintenance windows, changes within windows not correlated with expected maintenance activity.

Any post-install change in file permissions.

Routers: Changes in configuration outside of normal maintenance windows, changes within windows not correlated with expected maintenance activity.

Web Servers: Changes in Web server security configuration parameters such as root directory and ability to index.

Service:

Administrative Access: Unable to connect to systems or networks.

Dial-In Application Access: Busy signals or unanswered calls.

Internet Application Access: Ability to access application data without login. URLs not resolved or unavailable.

Outgoing Internet Access: URLs not resolved or unavailable.

Mail rejected.

System and Network Mgmt: Changes in the sets of protocols or ports served by system or network platforms

Ability to perform system management functions through unauthorized accounts or networks.

Intrusion

Defining ways to measure intrusions is much harder than defining ways to measure corruption. It first requires a definition of intrusion. To summarize the definition in Webster, an intrusion is an act of entering a place without invitation or welcome. To define an intrusion, one must first define the place. When one defines intrusion with respect to a house, does the place under scrutiny include the walk, the yard, the doorstep, as well as the building? If a stranger hung out on your front walk or back yard, would you call that stranger an intruder?

To extend the analogy to information systems, the place is the set of systems under scrutiny as well as the network connectivity constructed and maintained for the purpose of accessing the systems. The broadest definition of intrusion would include any activity on those access paths. For why should any stray packet enter your network if it does not have a legitimate purpose in accessing your systems? However, by our analogy, this stray packet is akin to a stranger standing on your driveway. We all understand that is it common for people to get lost, to be ignorant or mad, or to just be deliberately obnoxious. We all tolerate some level of trespass in allowing for such common behaviors.

Successful intrusion detection draws a clear line between those random acts of trespass and intentional unauthorized access. Almost all intrusion detection "packages" have customizable settings which allow you to define how many neighbors may randomly stand on your driveway before you label them a threatening mob. This threshold will obviously be vary with your address. As few as 10 people if your address is Potter's corner, and as many as 100 if your address is the corner of 6th and 45th. It is typical in planning an intrusion detection strategy that the focus starts on those numbers.

After all, it is an easy win to identify the traffic patterns that are normal for your neighborhood, aggregate them into normal traffic patterns, then look for huge spikes. This methodology will detect the undeniable denial of service attack. But the definition of an intrusion does not start with the numbers. The definition of an intrusion starts with the intention of the individuals. The numbers correlate with the intention of an angry mob, but they are not the cause. They are the effect. If the network traffic numbers are caused by a misconfigured system, they are not indicative of an intrusion. Successful intrusion detection will not stop and alert for unusual activity, it will follow the behavior of its object until it is determined that the activity is harmful or until the threat is relieved.

Acknowledged that privately constructed access paths will be trampled in intrusion and that if no intention of further exploiting such access is made, the trespasses do not count as an intrusion. While this traffic may be interesting and even worthy of investigation, it does not count as a security defect and is not relevant to measuring how well system are secured. Hence, the intrusions coincide with those cases where access paths are trampled and successfully exploited to gain unauthorized access. This is a definition for intrusion that provides a basis for a defect detection strategy.

With this definition, it is possible to detect intrusions based on access paths exploited (the defect) and intentions for further exploit (the evidence it is possible to gather). In the above example, the web server needs an access path from the Internet in order to let users into the application. That path includes the protocol and port allowed through the firewall, as well as a login and password on the web server itself. Say, upon login, there are 3 links the user can choose from and that the total service proffered by the web server is comprised of 4 URLs, the login screen plus the three links. An unwanted visitor may access the web server via the access path through the firewall and enter something into the web server other than one of those 4 URLs. As long as access is denied, there is no security mechanism broken in the attempt, there is no defect, and the activity does not constitute an intrusion. However, say that the visitor did not login but successfully access a URL that was one of the links past the login screen. Logs in this case should indicate that someone managed to bypass the application controls (a defect). This activity is an intrusion.

An event is an intrusion if it meets these two necessary and sufficient conditions:

- successful penetration of an access path
- attempts at exploiting access from the point of penetration

Given this definition of intrusion, some example metrics indicating intrusions in the security of this environment are:

Logs:

Application Servers:	Functions accessed out of menu sequence. Database access through direct queries rather than stored procedures. Same user logged in from multiple locations at same time.
Dial-In Servers:	IP address allocation log without corresponding authentication log.
Firewalls:	Administrative access outside of maintenance windows.
Monitors:	Logs that have decreased in size when they should only increase.
Operating Systems:	Logs of unauthorized users gaining superuser privileges. Starting and stopping of unauthorized services.
Network Probes:	Logs of unexpected protocols and ports in packets in protected areas of the network.
Web Servers:	URLs accessed that are not part of the application.

Configuration:

Application Servers:	Changes in access permissions that cannot be correlated with user administration activity.
Dial-In Servers:	Unexpected changes in administrative access rights.
Firewalls:	Unexpected changes in administrative access rights.
Monitors:	Unexpectedly altered inventory data.
Operating Systems:	Destructive commands outside of normal maintenance windows (e.g. system reboot).
Network Probes:	Interfaces disabled.
Routers:	Destructive commands outside of normal maintenance windows (e.g. system reboot).

Service:

Intrusions themselves will not be detected by testing services. Service testing can only indicate whether an intrusion is possible, given a configuration defect.

Failure Scenarios

Suppose that all security mechanisms are monitored for corruption and that as many intrusion scenarios as possible have been quantified and traps set for their detection. Could it ever be case where all automatic indicators tell us security is working, but we experience an intrusion. No lights or bells go off, but we find fraud in a user account? Does this indicate that the assumptions of the model are incorrect?

It certainly indicates that the process for measuring security must be changed. But it does not threaten the defect elimination model for measuring security unless the root cause of the fraud is identified and it is not possible to automatically monitor for it. If the root cause is identified and it can be added to the security measuring process, then the model remains a success. The security mechanism not previously thought of must be added to the pool and measured. The main reason it does not indicate a defect in the approach is that, as automatic and investigative approaches start with the same assumption of security objectives and risks, the corresponding investigative approach would also have missed the incident. Another possible flaw in the automated measurement approach is that it requires a successful intrusion in order to count a security defect. It may be presumed that the Security Metric Server will also be used for intrusion alerting. Incident response must be part of an effective security process, and any security metric model must not judge an IT environment secure unless it can facilitate appropriate incident response activity. Yet a security metric system that does not catch an intrusion until it is successful does not allow the organization to get an alert early enough to respond to intrusion threats.

Consider the example of distributed denial of service attacks. The defect elimination model of security metrics can handle this by setting the definition of defect as a bandwidth utilization threshold that is very high, but just lower than required to bring down routers and operating systems. The security metric does not have to be a defect in the sense of the word that something is wrongly configured. The defect in the distributed denial of service attack is that the bandwidth threshold penetration allows an exploit. A defect shared by all existing Internet systems is nevertheless a defect. Identifying it as a defect in the security profile leaves room in the model for continuous improvement. As new technologies emerge with which to address the problem, their implementation will lower the defect level.

Summary

This article has described current five models for measuring information security:

- External Audit
- Internal Audit
- Capability Maturity
- Risk Analysis
- Defect Elimination

These models were classified into two approaches:

- Investigative
- Automatic

The models and approaches were compared and contrasted. The defect elimination model in combination with the automatic approach was recommended.

It was described how this approach could be used to measure security defects. Security defects were categorized into two types:

- Corruption
- Intrusion

Mechanisms for measuring security defects of each type were described in detail. Concerns with possible flaws in the defect elimination model in combination with the automatic approach were observed and addressed.