# Global Information Assurance Certification Paper

# Getting a Handle on Security Events

GSEC Practical Assignment v.1.4b (Option 1)
By Sean T Murray
04/30/2003

***Abstract.***

One of the many problems facing the security teams of large enterprises is the detection of security incidents. This is due mainly to the large amount of data from security devices. A new breed of software solutions, being called SIM (Security Information Management) or SEM (Security Event Management) is emerging. This paper, whiling not delving into particular vendor solutions, outlines the advantages and caveats of SIM/SEM solutions.

***The Problem***

Security incident detection is dependent on security events. These events come from operating systems, firewalls, routers, Intrusion Detection System (IDS) sensors, virus software, E-mail scanners, and other miscellaneous security devices. Current terminology calls them alerts when they sent/processed (near) real time, and logs when they are stored for later review. Either way, they are the clues that tell when there has been security problem. These logs are in different formats, have different data fields and, in a large organization, could result in hundreds of gigabytes of data per day.

The problem is simple—information overload. In a large enterprise environment there is so much data, even from a single device that it is nearly impossible to make sense of it all. Reviewers need to detect and investigate the "important" events and ignore (although probably archive) the other events

Some specific problems of event review are:

- **Reviewers**. People who are doing event review (whether it is near real-time alerting or after-the-fact log review) have two dimensions of knowledge they must have:
    1. Computer Security—Obviously, the person must know what each event they are looking at means. This can often mean a person must know:
        - Operating systems and what they log
        - Networking protocols
        - Attack types
        - Attack signatures and how they would look in each data source (OS log, IDS log, etc)
    2. The Environment— The reviewer must know what are "normal" event patterns for the environment they are reviewing. For example, imagine

a reviewer, looking at Network IDS, sees a rash of different web attacks from an internal host against a Web Server. The reviewer, knowledgeable in computer security, will open an incident and get many other resources involved. A reviewer who knows the environment may have seen the pattern of alerts before, look at the source, and realize that this is an application developer running a vulnerability scan against a new application server. The reviewer then could make a quick phone call or do a lookup to configuration management software, get the necessary information and close the incident. This type of knowledge is difficult to transfer or even store effectively. Optimally, a reviewer would want to have the ability query or look-up "notes" created by reviewers on events that have been investigated in the past. This environmental specific knowledge is even magnified by the need for the reviewer to know :

  - Policies or rule sets on the various security devices in the environment.
  - Network topography of the environment.
  - Which hosts are important as far as data sensitivity, mission critical processes, etc.

- **Limited resources**. There is no way to investigate every event. Deciding what to investigate is critical.
- **Limited view**. An incident may not be apparent from a single event, but rather from multiple events from multiple sources. Cross-referencing events over different security devices is difficult, due to the volume and the non-conformity of data of the various devices.
- **Data storage**. Each device will have different formats for the events. The events need to be stored so they can be:
  - Retrieved quickly if an investigation warrants. An investigation of a security event may involve the firewall logs, router logs, network IDS, virus logs, host IDS, OS logs. Allowing the investigator to get the information from each of these is challenging. Even if the investigator knows where to look and the format of the log, some queries may take a long time (in the order of hours) if the logs are not stored correctly. There is so much data, either the data is stored in a single huge location, so the query takes a long time, or in many locations, so the investigator must do multiple queries and collate the data.
  - Archived for a period of time, according to the policies of the organization.
  - Retrieved from an archive, if the investigation is about an event that happened a long time ago.
  - Used in a court of law. If there is an investigation that leads to prosecution of an individual, the logs may be needed in court or other legal action.

***Solution- Get Software to Help***

There is a new breed of software that is being called SIM (Security Information Management) or SEM (Security Event Management). Some functions of SIM/SEM software are (Mostly taken from [1]):

- Collect from a large variety of security devices and systems.
- Correlate security events. This takes individual events from different sources and combines them to give the reviewer more information about a possible incident. Correlation can either be automatic (based on rules) or run ad hoc by the reviewer.
- Classify resources. Alerts generally show IP addresses or machine names. Classifying resources uses these to display more information on the devices in the alert. This allows a reviewer to immediately see what the hosts in an event are, where they are physically located (both inside the network and outside), if they are high risk, contain sensitive data, etc.
- Normalize the data. Each security device has a different format and different fields of data collected. Normalization involves taking the raw alert and putting the data into a single format.
- Aggregate the data. (This is similar to Correlate.) Aggregation is the process of taking individual alerts that are part of a single event and combining them to decrease the volume of the data needing review.
- Store the data in a database. This includes storing the normalized data (for review and reports) and the original raw data (in case information is lost in normalization and also to provide a mechanism to gather evidence for legal purposes).
- Filter or create alarms based on various (complicated) criteria.
- Create reports
- Create a knowledge base of an organization's individual event patterns and past investigations.

The rest of this paper will outline each of these, and described both their benefits and their caveats. There is an appendix at the end, which lists various vendors in the SIM/SEM space. This paper will not look at individual software that does SIM/SEM, but rather will try to help the reader understand what they may want or need in SIM/SEM software and to understand the limitations of each benefit.

### *Collect the Data*

There are several issues about getting the data from the source (security device, computer host, etc) to the SIM/SEM system.

1. **Deciding what events to capture at the source**. One way to limit information overflow is to limit the information. Of course, we don't want to miss important events that indicate a security incident. For example, most unsuccessful logins to a host are simply a user mis-typing their username or

password. Similarly, most drops from an Internet firewall are probably the latest worm/virus/Trojan trying to blindly infect the network. The data must be captured for analysis, even though most of it "uninteresting". The first place to start tweaking the data is the policy or rule set on the devices generating the events. The general methodology is to start with a configuration that may be "close" to optimal, analyze the data, and begin to tweak the configuration. If a rule or alert generates too much information to be useful, then it must be altered or filtered at the source (that is, never sent to the central data store). A SIM/SEM solution can help to manage the rule sets or policies on devices by the reporting and visualization mechanisms (see below), which can help weed out rules or alerts that are redundant or generating too much data.

For some guidance

- A good general Log Analysis resource with many links is http://www.counterpane.com/log-analysis.html:
- Computer Operating Systems Logs. Each operating system varies on what can and should be logs. For some guidance see The Sans Consensus Guides (http://store.sans.org/store_category.php?category=consguides ), or The Center for Internet Security Benchmarks (http://www.cisecurity.com)
- Intrusion Detection System (IDS) alerts are more problematic. Each IDS solution has different alert signatures and each environment will have different alert patterns. Then, just when everything is configured, new signatures for the IDS are created and they must be incorporated into the sensors. For guidance check the documentation of the individual IDS. For general information see the snort homepage (http://www.snort.org) or see [2]
- Firewalls and routers. According to [3], normally logs are based on each rule and generally packets that are permitted are not logged. The security professional must assess each "deny" rule and make a decision whether to log when the rule fires.

2. **Scalability**. When dealing with the amount of information that a SIM/SEM solution can potentially process, it may be impossible to have the processing occur on a single machine. The SIM/SEM architects/administrators may have to break up the processing among machines. A successful SIM/SEM design must be able to do this. Some SIM/SEM software allows the creation of a hierarchy of SIM/SEM processors and databases that can be divided any way that make sense. For example, an organization can have some people doing just IDS review, others doing firewall review, others doing OS review, and so on. If a reviewer sees something odd or beyond their experience, they can pass the event(s) on up to a higher processor in the hierarchy, to be reviewed by second level reviewers. Alternatively, an organization can segregate geographically, with people at each site reviewing alerts created at their site. This has the advantage of keeping the environmental knowledge (network layout, important applications/servers, traffic patterns, etc) of the reviewers local, where they can keep on top of changes. Again, these reviewers could

send “odd” alerts on up to a second level processor. These solutions could also be combined. The first-line processors could also automatically (based on rules) send certain events to the next level processor. Flexibility is the key to the scalability of the SEM/SIM processing.

3. **Getting the data from the security software to the SEM/SIM software**. The SEM/SIM software must be able to receive the data from the source device creating the events. A key requirement when selecting SEM/SIM solution is to transport the information confidentially and reliably. There are several ways to get events from the native device to the central system. Some standards are:
   - Syslog. This is the Unix protocol that has become a standard for system logs. (see [4],[5])
   - SNMP. Simple Network Management Protocol.
   - File transfer for logs (ftp, ssh, network shares, etc)
   - OPSEC (Open Platform For Security)—Checkpoint Software’s standard for security devices and integrated security solutions (See [6])

   If a security device doesn’t use one of the above transports, the following mechanisms can be used:
   - Run an agent on the security device that uses either a proprietary method or one of the methods above. This is necessary for the Microsoft Windows Operating Systems (see [7]). Agents can also act as filters to limit unwanted events from being sent to the central processor(s). They can also take some of the processing burden off the central SEM/SIM servers (at the cost of adding processing burden to the device). The Agents are usually part of the SIM/SEM package.
   - Run an agent that extracts from the native security software’s database or log file, and then sends the data. This usually involves an agent polling the data source for a change and passing the changes to the SEM/SIM processor.

   Some transport issues that have to be dealt with are
   - Volume—This can create a significant network load, especially during an attack or virus/worm outbreak.
   - Authentication— Does the event collection facility authenticate who is sending it data? Could someone send false information to confuse the reviewer?
   - Reliability—Syslog and SNMP use UDP as a transport. UDP is unreliable, so there is no way to be sure that the data was received properly.
   - Encryption—The events will be traversing the network, and possibly some untrusted network segments. It’s a good idea to encrypt this data, since it will contain a lot of information about the enterprise’s network, hosts, vulnerabilities, and security software.

   If the native transport mechanism doesn’t provide these, then an add-on must be used to protect the data in transit. Some examples of securing the traffic

are Virtual Private Networks (VPN) (see http://www.vpnc.org), IPSEC (see http://www.ietf.org/html.charters/ipsec-charter.html ), or an SSH tunnel (see [8]). There is work being done to add security to syslog in the form of an IETF Internet-draft on Syslog-Sign Protocol. (see [9])

***Normalize the data.***

One of the biggest obstacles when dealing with different security devices is that each has a different alerting format. Normalization is the process of taking each native event and transforming it into a pre-determined format. This involves taking the native data points (source IP, source port, destination IP, destination port, date/time, machine name, user name, etc) and filling it into a common structure. XML is the most convenient and seem to be emerging as the standard structure language for the normalized data. When a reviewer looks at the data, they only have to know the data points in the normalized form. They can quickly look at the pertinent information without having to know the field layout or field names of the native alerting system.

Not every alert will fill all data points in the post-normalized format. For example, firewall events may or may not have usernames, but logon events will always have username.

The trickier aspect of normalization is putting the event type into a common taxonomy or language. An organization's IDS, firewalls, Operating System and other security devices may send hundreds types of alerts. Many of these alerts are semantically identical or may fall into the same category. For example, a TCP port scan alert from an IDS is semantically the same as the hundreds of drops from a single host to a single host over many ports, as reported from a PIX firewall. A logon event from Unix, is semantically the same as a logon event from NT or any device.

Turning the native type into a common dictionary or taxonomy of events is difficult because there are no standards (yet). The situation may improve. There is an IETF Working Group, named Intrusion Detection Exchange Format (idwg) (http://www.ietf.org/html.charters/idwg-charter.html) that is working on this. Currently, each SIM/SEM vendor has it's own standard. Although it can usually be tweaked, that is what an organization is going to have to live with. When selecting SEM/SIM software it is important to find out how the vendor came up with it's taxonomy and whether it fits within an organization's view of security.

Normalization is necessary for most of the other advantages of SIM/SEM, so it must be done correctly to get the most benefits from a solution.

***Correlate security events***.

Correlation is the cross-referencing of individual events from different sources to give the reviewer more information about a possible incident. A good example of how correlation can be useful is with login failures. A single login failure to a system (single event from OS log) may not warrant investigation but 10 failures on 10 systems for a single login name within 5 minute would warrant further analysis.

The key to correlation is timeliness and relevance (see[10]). To achieve the timeliness, the correlation engine must process a great amount of data quickly, so it requires a great deal of processing power. Care must be taken to architect the SIM/SEM solution with enough processing power to handle the load, especially during a "noisy" event, such as a worm or virus breakout.

Correlation can be done either:

- Automatic- The reviewer sees all the pertinent information with no effort
- Ad hoc- The reviewer can correlate with a few mouse clicks and/or keystrokes.

A downside of automatic correlation is deciding the relevant data to correlate. Correlation is based on rules and the creation of correlation rules is not trivial. Without normalization to a common dictionary (see above), automated correlation isn't that useful (as explained in [11]). This is because correlation only makes sense when the system "understands" an event, and can go out to find only the data about the event that is relevant. Haphazard correlation will only muddle the picture. A system cannot correlate everything, so without some base semantic "knowledge" of events, automated correlation only adds data. Therefore, any organization looking at SIM/SEM must realize that automatic correlation will have some administrative overhead in the creation and update of correlation rules.

***Aggregate security events***

Aggregation is the process of taking many events and turning them into a single alert. A single event may result in the creation of many alerts and log entries from multiple devices. Aggregation tries to reduce the data by showing the reviewer the single alert, instead of all the events that were actually created. Aggregation is similar to correlation, in that each:

- Looks across alerts and log entries to try to find related data
- Is based on rules that must be defined
- Depends on normalization of the data
- Can be processing intensive when there are large amounts of data

This means that aggregation comes with the same caveats as correlation (see above)

***Classify resources.***

Alerts often show IP addresses or machine names. The SIM/SEM solution can classify resources so a reviewer can immediately tell whether a resource is high risk and whether an immediate investigation is warranted.

Classification dimensions include:

- Logical location (backend, DMZ, extranet, Internet, etc)
- Physical location
- Classification of data stored on the machine/subnet
- Importance of availability of machine/subnet
- Patch level of the OS on a host

The reviewer can benefit from knowledge of these dimensions when responding to events on a device. Viewing this data as part of the event itself is very helpful to speed up the decision process of how much resources should be spent investigating an event. The larger the environment, the greater an advantage this is. There is, however, no free lunch. There is, of course, the initial loading and the updating of the classification scheme.

It is important to note that the classification is usually based on IP and the classification rules may work at different levels. If the network is sub-netted properly, classification may be based on sub-net information instead of the full IP address. Some solutions allow the classification to first to classify based on the full IP (that is, the host). If there is no match there, the system will try it at each subnet level until a match is found. (If no match is found, the classification would be “unknown”)

***Store the data.***

Storage includes storing the normalized data (for review and reports) and may also include storing the original raw data. The raw data may be needed when information is lost in normalization and also to provide a mechanism to gather evidence for legal purposes. Since this will store redundant data (in raw form and in normalized form), more storage space is needed, adding to the already demanding storage requirements.

The problem is the amount of data in the storage unit. When ids, firewall, and OS logs are included, a medium or large enterprise can have hundreds of thousands or millions of events in a day. Even on a large enterprise class server, a database quickly gets slow query response time.

Possible solutions to this are:

- Store in multiple databases or tables. The data could be stored in different areas based on priority, source of the event (physical or logical), or other criteria. This, of course, severely limits the gains of having the data available for reporting and querying in a single location. Another separation method would be to put archive data (data that is thought to be not critical, but may be necessary if an investigation has to be done) into a single database. Then the other events (which would be anomalous activity) would be put into another. This way the reviewers can quickly get at the anomalous ("interesting") data and keep it online for a longer time.
- Archive and purge the data often. This means keeping the data online for a shorter period of time. This may be fine, as long as the archive data is retrievable in a format that can be of use.

Any SIM/SEM solution must come along with tools and procedures to allow the administration of the data, including querying, archiving, and purging.

***Filter and create alarms based on various criteria.***

Each data source has various types of events that reviewers may want to ignore:

- False Positives from IDS, e-mail filters, virus software and other rule- or signature-based security software. The SIM/SEM can filter alerting of events that are known false positives, but cannot filter via the native security software.
- Data stored just for archival purposes. Some events may be captured just for archival purposes- because organizational policies say so (for legal purposes) or they may be needed in case of an investigation.

SIM/SEM should give the ability to allow of this data to be ignored by reviewers, but still stored in a database for archival purposes and available to the correlation.

Using the other features, a SIM/SEM solution can also create alerts based on multiple and complicated criteria. For example, an organization may want to be alerted if there are 10 failed logins using the same user id to one or more machines in a 5 minute time period. Classification also helps when creating or upgrading alerts, because the alerting rules can look at the sensitivity or importance of the host in the original event. The SIM/SEM solution could also send an alert over various channels:

- Reviewer's console
- E-mail
- Page
- SNMP or other monitoring alert

### *Create Reports and Visualizations*

Arguably, the greatest advantage of consolidating all security events is the reporting. This is where SIM/SEM can really shine for helping to get a handle on the state of the enterprise from a security standpoint. As a bonus, management can get all the data they wish and see that their security investment is being put to use.

Criteria to look for in reporting are along the same lines as any reporting tool:

- What is needed on the front end to run reports (web browser or client software that must be installed)?
- Are there licensing or technological limits to the number of concurrent reporting clients?
- How many canned reports come with the solution?
- Are reports easily created, customized, printed, and exported?
- What kind security is attached to the reporting mechanism, so that only permitted users can access the reports?
- Can reports get saved and shared easily?

Visualizations are another reward of consolidating security data. Visualization takes the normalized, correlated data and displays it on the screen as 2 or 3 dimensional graphs, trees and "information spaces". One can see information based on source, destination, ports, event, and time. Then, at a glance, the reviewer can see the state of the network. The user can also drill-down or otherwise navigate the information space, to see anomalies in the data that would be difficult to see in a report, but stand out when visualized.

Visualizations are slick and can be very useful, but one must be careful to understand how they will fit into the review environment. The following questions must be answered before visualizations can be successfully implemented into a solution:

- Which visualizations are useful? Visualizations allowing the user to view the data in many different ways. Which dimensions (source/destination IP, time, event name, machine classification) make the best visualization? The goal is to save time by having the ability to either see the state of the network and to find anomalous data. Does a particular visualization accomplish that?
- Once it is determined what visualizations are useful, how often will they be used? Will someone be monitoring the visualization 24/7? Will someone do information mining visually with stored data once a day, once an hour, or other time period? Does any useful information come out of the visualization or is it basically just more data or noise to purge through?

### *Create a knowledge base of an environment's individual event patterns.*

For the reviewer, knowing which events are "normal" traffic for their network is a must. An investigator often spends time researching an IDS alert, only to find

that it is legitimate traffic that only occurs sporadically. How do investigators spread that knowledge to other reviewers (not to mention remember it when they see the alert again in the future)?

SIM/SEM offer a few solutions to this problem:

- Some allow the users to create a knowledge base of alerts with comments. This allows investigators to query to see if anyone has seen this type of traffic before.
- Some allow the users to add rules that add notes to events. When the event is seen again, notes or comments will appear to the reviewer, so they can immediately know if a similar event has been investigated before. The rules could also:
  - Hide the event (so the reviewer never even sees it)
  - Change an event's priority
  - Control how (or if) the event eventually get stored.

### *Other functions*

Integrating a SIM or SEM solution with vulnerability scanning and tracking software is advantageous, because this allows the user to track two dimensions of the risk management equation – vulnerability and threat. The SIM/SEM can help show the threat and the vulnerability tracking software can show the vulnerability.

The SIM/SEM solution must also have internal security. There will be people who are required to change correlation, normalization, aggregation and the other rules in the system. There will be people who can view reports, possibly only a certain subset of reports. A SIM/SEM solution must be able to allow the management of users, passwords and rights. A scalable solution would also add roles so that rights could be mapped to users and groups of users.

### *Conclusion*

SIM/SEM solutions can help a large organization detect and react to security incidents by organizing the events from security devices and hosts. However, when implementing a solution each function of the solution must be analyzed and executed properly to ensure that the solution is effective and manageable.

## Appendix A— Quick List of Items to look for in a SIM/SEM solution

- The number of security devices and agents from which the solution will accept and parse logs and alerts.
- Ease of adding events from unsupported security devices and agents.
- Scalability.
- Ease of use
  - Managing correlation and aggregation rules
  - Managing classification
  - Managing normalization
  - Performing database maintenance
- Reporting
  - Number of clients
  - Client software needed
  - Number of canned reports
  - Ease of use creating, modifying, and exporting reports
- Other functions the solution can do.
- The different channels the solution can use to forward or escalate alerts.
- Internal security—maintaining users and user rights

## Appendix B—Short List of SEM/SIM Software Vendors

ArcSight
http://www.arcsight.com

eSecurity
www.esecurityinc.com

GFI LanGuard
http://www.gfi.com/languard/index.html

Intelletactics Network Security Manager (NSM)
http://www.intellitactics.com/products/nsm_overview.html

LogSmart (with Network Intelligence Engine hardware)
http://www.network-intelligence.com

NetIq Security Manager
http://www.netiq.com/products/sm/default.asp

Netforensics
http://www.netforensics.com

NeuSecure
http://www.guarded.net

SilentRunner
www.silentrunner.com

## References


[1] Dubie, Denise. “Users Shoring Up Net Security With SIM”. Network World. 30 September 2002, URL: http://www.nwfusion.com/news/2002/0930apps.html (30 April 2003)

[2] Bace,Rebecca and Mell, Peter. “NIST Special Publication on Intrusion Detection Systems.” http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf (30 April 2003)

[3] - CERT. “Configure Firewall Logging and Alert Mechanisms.” CERT Security Improvement Modules. 1 May 2001. URL: http://www.cert.org/security-improvement/practices/p059.html (30 April 2003)

[4] Lonvick, C. “The BSD syslog Protocol.” IETF Request for Comments: Category: Informational” August 2001 . URL: http://www.ietf.org/rfc/rfc3164.txt (30 April 2003)

[5] Pitts, Donald. “Log Consolidation with syslog.” Sans InfoSec Reading Room. 23 December 2000 URL: http://www.sans.org/rr/unix/syslog.php (30 April 2003)

[6] Checkpoint Software Technologies LTD. “Intro to OPSEC Integrated Applications.“ URL: http://www.opsec.com/intro/applications.html (30 April 2003)

[7] Garbrecht, Frederick. “Practical Implementation of Syslog in Mixed Windows Environments for Secure Centralized Audit Logging.” July 17, 2002. URL: http://www.sans.org/rr/casestudies/mixed_win.php (30 April 2003)

[8] Chuvakin, Anton. “Advanced Log Processing.” Security Focus Infocus. 1 August 2002. URL: http://online.securityfocus.com/infocus/1613 (30 April 2003)

[9] Callas, J. and Kelsey,J. “Syslog-Sign Protocol .“ IETF Internet-Draft syslog Working Group. February 23, 2003 URL: http://www.ietf.org/internet-drafts/draft-ietf-syslog-sign-10.txt (30 April 2003)

[10] Hollows, Phil. “Security Threat Correlation: The Next Battlefield.” ESecurity Planet. 14 November 2002. URL: http://www.esecurityplanet.com/views/article.php/10752_1501001 (30 April 2003)

[11] Chuvakin, Anton. “Cross Platform Security Analysis.” The Internet Security Conference Newsletter. Volume 4, Issue 18. 13 December 2002. URL: http://www.tisc2002.com/newsletters/418.html (30 April 2003)