

[**Editor's Note:** The following excerpt is from the free eBook *The Definitive Guide to Security Management* (Realtimepublishers.com) written by Dan Sullivan and available at <http://www3.ca.com/ebook/default.aspx?sacid=60453>.]

## Chapter 4: Security Risk Management

Information security professionals are rarely at a loss for data. Point products—such as firewalls, intrusion prevention systems, antivirus programs, operating systems (OSs) and other elements of the security infrastructure—generate steady streams of data about events and conditions. Security professionals are not in need of data—they need information. Thousands of events in an intrusion prevention system may be triggered by a single incident. Hundreds of events logged by a firewall might be related to a single distributed denial of service (DDoS) attack. Seemingly separate events identified by firewalls, intrusion prevention systems, and antivirus programs may all relate back to a single attack or incident. Filtering volumes of raw data, correlating events, and reporting actionable information in these situations is the role of a security information management (SIM) system.

### What is SIM?

SIM is a framework for organizing, analyzing, storing, and using security event data. SIM serves several purposes:

- Real-time monitoring
- Post-event investigations and forensics
- Effective communications
- Security planning
- Compliance regulation
- Disparate security system management

Each of these purposes depends on correlated and aggregated security information that is generally not available from point systems alone. Even when it is available *within* a given system, it is not readily shareable among other security point systems.

### **Real-Time Monitoring**

Real-time monitoring is the ability to identify security events as they happen, assess the potential threat to the organization, and support appropriate responses. When fast-spreading malware, such as the SQL Slammer worm, strikes a network, alarms are triggered, logs fill with detailed data, and point products such as antivirus programs react. The first task for security professionals at this point of an attack is to rapidly assess the potential impact of the threat and identify high-priority resources that must be protected.

SIM aids this process by aggregating and normalizing data. Are the 100 alarms the result of 100 separate incidents or one incident? Can concurrent attacks be sorted when monitoring software is scrolling events faster than they can be read? SIM helps organize raw data around logical events and reduce the flood of information. It can also help focus and prioritize responses by providing answers to questions such as:

- Are some alarms triggered on machines that have already been patched for a given threat (meaning that they're producing false positives)?
- Which are the most important of the vulnerable systems?

SIM can also integrate vulnerability management information, which is essential to understanding the potential impact of a threat.

### ***Post-Event Investigations and Forensics***

The same data that drives real-time monitoring is also needed after an attack. Post-event investigations and forensics depend on audit information and other evidence for two reasons: to analyze the detailed nature of an attack and gather evidence for legal proceedings.

Unlike real-time monitoring, which is designed to prevent and suppress attacks as they occur, forensic analysis is done after the incident is under control. Key management issues in forensics include:

- Ensuring that the stored data, after aggregation, maintains a record of all events
- Preserving a chain of custody of evidence
- Maintaining integrity (preventing the tampering of audit data)
- Correlating information from multiple point systems to fully understand the nature of the attack

Forensic information may be used in civil or criminal proceedings, so procedures in addition to typical standard operating procedures (SOPs) may be required to meet legal standards for preserving evidence.

### ***Effective Communications and Risk Management***

Security management depends on the close involvement of both technical and business professionals. C-level executives, auditors, HR personnel, and others need to understand the nature of threats and their potential impact on an organization in terms and with views that make sense for their roles. In especially serious situations, executives may need to make decisions about shutting down core IT services. SIM systems are needed to provide an integrated, high-level picture of security events so that decision makers can quickly determine how to respond, both from a technical and business perspective.

SIM supports long-term communication objectives as well. All users, from data-entry clerks to executives, should be aware of basic security policies and procedures. When properly presented, information gathered with SIM systems can help users understand the speed at which threats can spread, the impact they can have, and scope of remediation required to recover.

## Security Planning

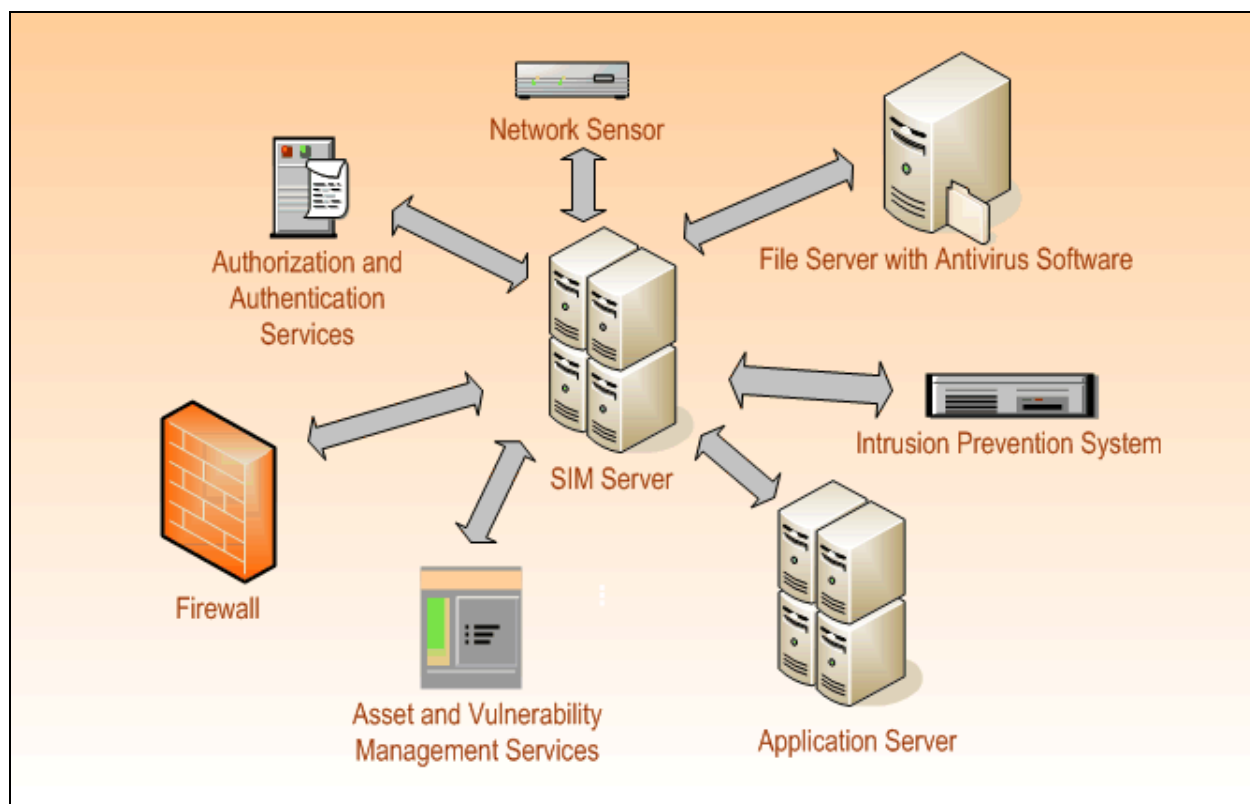
Organizations define roles and responsibilities, response protocols, and recovery procedures to deal with security incidents. The SIM framework allows enterprises to leverage SIM functionality with respect to these responsibilities, protocols, and procedures. Alerts and reports can be sent to those in specific roles, other reports can provide correlated network information needed to respond to an attack, and audit trails and forensics reports can pinpoint resources that were affected by the attack.

Unlike other security systems that support preventive measures, SIM focuses on managing security risks so that the business is protected and targets remediation—thwarting an attack when it occurs and collecting and correlating information to support post-event analysis. This post-event information in turn allows an organization to improve its security posture.

SIM systems are analogous to business intelligence systems for executives. SIM systems collect, normalize, and aggregate large volumes of data and present essential information needed to respond to events. The processes for populating and using these systems are similar as well.

## Integrating Security Information

Like business intelligence systems, SIM systems depend upon other applications for raw data. As Figure 4.1 shows, SIM systems collect data from multiple systems and multiple types of systems.



**Figure 4.1:** SIM systems are repositories for integrated security information derived from multiple sources of security data.

### Dynamic Security Data

Dynamic data is gathered from security point systems, such as intrusion prevention, content filtering, and antivirus services. This data is typically generated in real time, as packets move through the network or as programs execute on a server.

OSs, firewalls, and applications write data to log files that describe specific events. For example, firewalls may record data about all rejected packets (recording a log of all accepted packets generates too much data for most organizations), authentication and authorization systems record events related to access controls, and applications log details of significant events in the execution of a program.

Sensors are devices placed throughout the network to monitor network traffic and identify suspicious events. Figure 4.2 shows a simple scheme for placing sensors in different zones within a network. These sensors can span multiple time zones, so it is essential for SIM systems to preserve sensor timestamps as well as standardize times to a global standard.

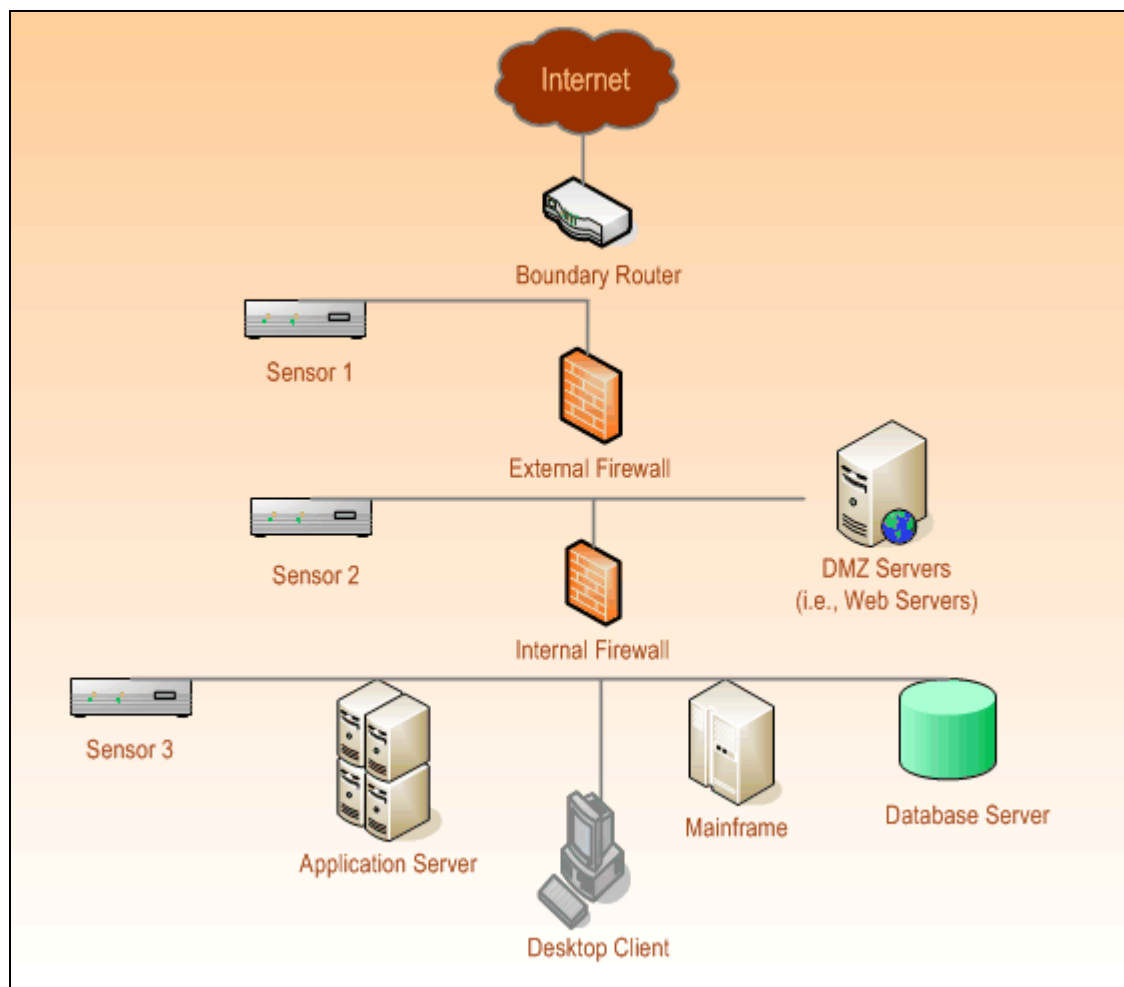


Figure 4.2: Sensors placed throughout the network can detect anomalous events in different zones.

Sensor 1, outside the external firewall, monitors the perimeter. This sensor has access to all traffic before it is filtered by the firewall, so this sensor can detect the level of probes and other security threats that launch against the network before security measures occur. Sensor 1 represents an opportunity to tell when someone is probing or fingerprinting a network for later correlation with data.


Sensor 2 is inside the DMZ, the zone between the internal and external firewall. The DMZ is used for application servers that are accessible from the Internet, such as Web servers. Sensors in the DMZ monitor activity that threatens these servers. Combined with sensors on the perimeter, sensors in the DMZ can track the effectiveness of the external firewall at blocking threatening traffic.

Sensor 3 is inside the internal network and monitors traffic that reaches the trusted part of the network. In an ideal world, there would be no threatening traffic in this zone; in reality, vulnerabilities can exist at any point in the network. Effectively managing vulnerabilities and patching critical servers and applications can reduce the likelihood of a breach in the internal zone.

A ranking of servers based on asset value can help to focus responses to malicious activity. Although the initial emphasis will be on protecting high-priority assets, non-critical servers can be compromised and used to stage later phases of an attack. In such cases, correlations between current and baseline activity can reveal unusual network communications. By itself, this information is insufficient to assess the impact of an active threat—organizational information is needed as well.

### **Organizational Security Data**

Organizational information describes the IT infrastructure of an organization. It includes asset, patch, and authorization information. The combination of dynamic and organizational information provides basic information about active threats, levels of vulnerabilities across assets, and the relative importance of various assets. SIM systems face several hurdles to integrate these data sources.

 For more information about asset and vulnerability management, see Chapter 3.

### **Challenges to Integration**

Key data integration challenges include:

- Normalizing network traffic
- Correlating events
- Meeting performance demands
- Ensuring data integrity
- Getting business intelligence out of raw volumes of data

## Normalizing Network Traffic

Sufficiently sophisticated attackers can avoid detection by network intrusion/prevention systems by taking advantages of ambiguities and complexities of Internet protocols. Scrubbing network packets to filter or correct the problem packets is called normalization. The need to normalize data is a result of the limits of network monitoring systems. Network monitoring systems might not account for all possible ways of transmitting a data stream through a protocol. Fragmented IP packets are resequenced when they reach the target system, but network monitoring systems may not account for fragmentation and could miss anomalous traffic patterns.

A network monitor can not always determine what packets are actually received by the target device. For example, a packet with a low time to live (TTL) count (the number of servers a packet can go through to reach its final destination) may never reach the target device. Attackers could send a stream of packets that appear, to the monitor, to be going to the target device when in fact only a subset of the actual stream reaches the target device. Thus, an attack can be hidden in an innocuous looking stream of packets. Detecting anomalous traffic can require contextual knowledge that is not available to a network monitor.

Scrubbing the network data can improve the analysis performed by a SIM system. Cleaner data yields better results. However, this process comes at a price, including an impact on performance and the potential to change the semantics of the data. There is also the slim chance that the normalizer can become the target of an attack. This situation is unlikely because the cost of breaching the normalizer is so high relative to its value to a hacker. To do so requires

- Knowledge of what point product is recording data
- Knowledge of what SIM solution is using data
- An understanding of how to exploit both of these in sequence

## Correlating Events

Once raw data is mapped to a common representation scheme, the next step is to identify relationships between events described by different systems. Some of the typical relationships include:

- Temporal
- Causal
- Equivalence
- Root-cause analysis

In many ways, temporal relationships are the easiest to identify; timestamps are commonly tracked along with other details in point systems. One problem that can complicate the process is the synchronization of clocks on different servers.

For most IT processes, slight differences in time tracking have no material affect on operations. SIM is a different story. Worms and viruses can spread rapidly and having precise and accurate timestamps across servers is essential to understanding the spread of malware. For example, suppose that a worm infects server A, which spreads to server B, then to server C. If the timestamps of alarms were 12:00:01.01, 12:03.02.01, and 12:00.01.80, respectively, it would appear that the attack began with server A, spread to server C, then to server B.

To some extent, the problem with time synchronization is avoided when data is aggregated. Grouping large numbers of events, for example 1000 individual port scans, into a single logical incident over a period of time reduces the need for precise timestamps. A large number of scans might indicate a reconnaissance phase of an attack. If those scans are followed shortly thereafter by an attempt to identify the OS, that is more evidence of an attack following a typical sequence of events regardless of the exact timestamps.

Sequential relationships are useful for defining rules to identify attack patterns. A SIM knowledge base may have rules such as *If a port scan on a server is followed by an OS fingerprinting operation, then issue an alert that a reconnaissance operation is underway.*

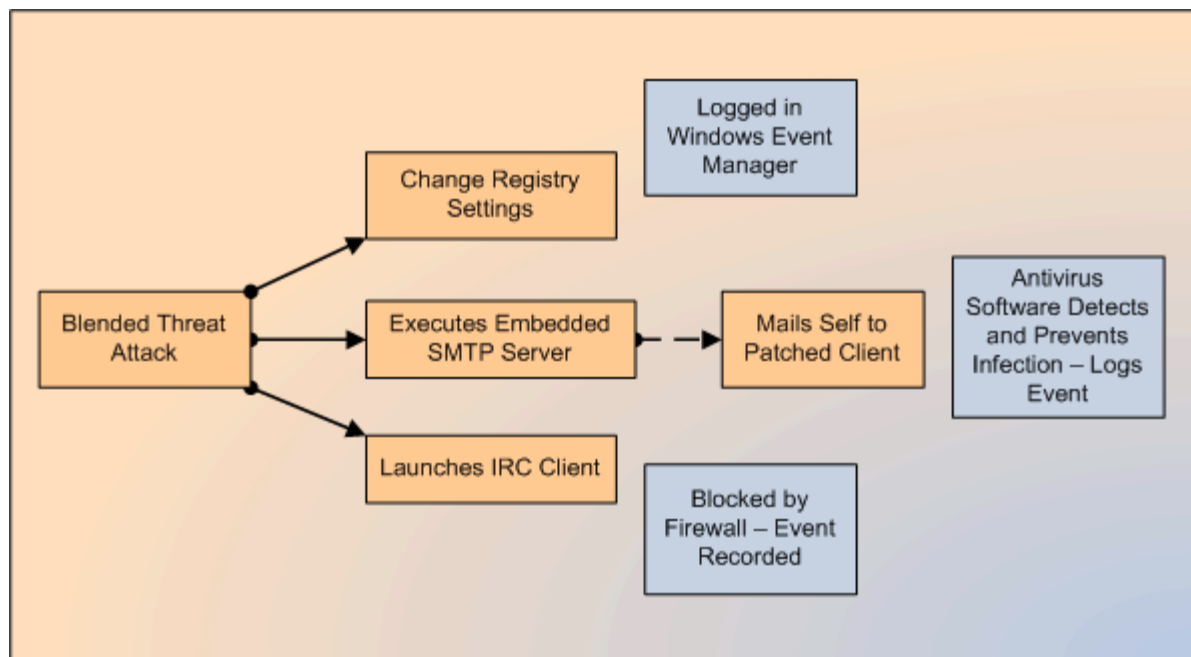
As malware becomes more complex, so does the problem of detecting related events. Consider blended threats that combine characteristics of viruses, worms, Trojan Horse, and other malicious code. These programs spread by several means and attack using several methods.

Nimda, for example, initially spread by email by exploiting a flaw in Microsoft Outlook that allowed the worm to spread without a user opening an infected attachment. The payload then infected file shares and IIS Web servers. From there, it exploited a bug in Microsoft Explorer, and spread to other clients. Other blended threats launch Internet chat programs to listen for further instructions, install keystroke capture programs, and attempt to compromise security settings. This sequence of events can happen in any order depending on the vulnerabilities encountered.

A single blended threat can trigger a variety of events in any order (see Figure 4.3):

- Malware attempts to download data or instructions from a known malicious site and content filters block access
- IDS records the detection of the malware's footprint on the network
- Antivirus software detects the presence of a worm in an email
- Firewall logs an attempt to use one of the Internet Relay Chat (IRC) ports (6665-6669)
- Windows Server event logs record changes to registry settings

Correlating these events requires that the SIM system have a knowledge base that describes threats and event patterns. That knowledge base is also useful for a third integration correlation challenge, equivalence of events.



**Figure 4.3:** A single event, the blended threat attack on an unpatched client, is casually related to four other events. Understanding those four requires understanding the causal relationship depict above.

In complex networks, a single event can have a ripple effect that generates a large number of event alarms. These are often called *alarm storms*. For example, if a router is under a DoS attack, the router may not respond to servers on the internal network. Each of these servers then generates an alarm indicating lost connectivity with the router. The servers may repeatedly test connectivity with the router, thus generate a repeating stream of alarms.

From a security management perspective, the repeating alarms from a single server can be aggregated into a single logical event with a time span ranging from the first failed connectivity test to the time communication is finally restored. A further level of aggregation would combine those logical events into a single incident: the router is failing to communicate with internal servers. Two procedures are used to extract patterns from raw data:

- Reducing data detects patterns among events themselves
- Aggregating data summarizes data by using a knowledge base of patterns

#### **The Need for a Knowledge Base**


Correlating security data into logically related events requires information outside that available in network traffic and security logs. A SIM knowledge base will include information about attack patterns, how alerts relate to one another, and heuristics, or rules of thumb, for interpreting events. Relations between alerts and heuristics for interpreting events are generally described by using if/then rules, also known as production rules. Attack patterns are described by both statistical patterns and production rules.

**Production Rules**

Production rules are basically if/then clauses that describe a set of conditions or events and a set of conditions or states that follow the previous conditions or events. The following example highlights one rule for detecting an FTP attack:

FTP Delete Rule:

If there is an ftpevent AND  
     The ftpuser is ‘anonymous’ AND  
     The ftpcommand is ‘DELETE’ AND  
     The ftpreply is ‘success’  
 Then  
     Issue an alert—ftp attack

 This example is taken from the P-Best research system on intrusion detection. The rule has been reformatted for readability. For more about P-Best, see “Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)” at <http://www.ce.chalmers.se/old/staff/ulfl/pubs/sp99lp-slides/SP99-PBEST-May99.PPT> and “Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances” at <http://www.sdl.sri.com/projects/emerald/project.html>.

Some systems will actually start a “state variable” given any one indicator and will then begin looking for a threshold or particular pattern from any subsequent one. This response doesn’t affect the previous FTP example, but it does with attacks in which one or two indicators are harmless but three means an attack.

There are several advantages to using production rules in the knowledge base. First, production rules are useful for making inferences and thus aggregate events. In the FTP example, a combination of several program states and events were reduced to a single event: an FTP attack. Second, production rules can trigger responses to anomalous events, such as shutting down a network session or blocking a port. In addition, new production rules can be added to the knowledge base without requiring changes to other rules in the knowledge base. Finally, production rules can describe facts that are independent of network activity—such as “tftp is a udp service.” These facts are useful for writing rules and can apply to a group of services or events—such as all UDP services—without creating individual rules for each member of the group.

However, there are also drawbacks:

- Production rules, as originally developed for use in artificial intelligence, are too slow for real-time network monitoring. Specialized rules engines, such as P-BEST, emphasize speed over other features.
- It can be challenging to identify all product rules that are needed for widespread security.
- Multiple rules can apply to a single set of circumstances. Rules engines use conflict resolution algorithms to select a single rule to execute. Developers must keep the conflict resolution algorithm in mind when crafting rules.
- Production rules and similar knowledge representation schemes can describe complex combinations of conditions. In some cases, basic pattern matching is sufficient.

### ***Attack Signatures***

Signatures are patterns that describe packets in a session stream. A security event occurs when a sequence of packets match the pattern. This method is popular in intrusion detection and prevention systems. Its advantages include:

- Speed
- Low alarm rates (with sufficiently detailed rules)
- Flexibility—new rules can be added as new threats are discovered
- Support for descriptive logs

The primary disadvantage of attack signatures is that they depend on known attacks. They cannot automatically adapt to new threats. A third method, statistical analysis, can overcome some of the shortcomings of attack signatures.

### ***Statistical Pattern Recognition***

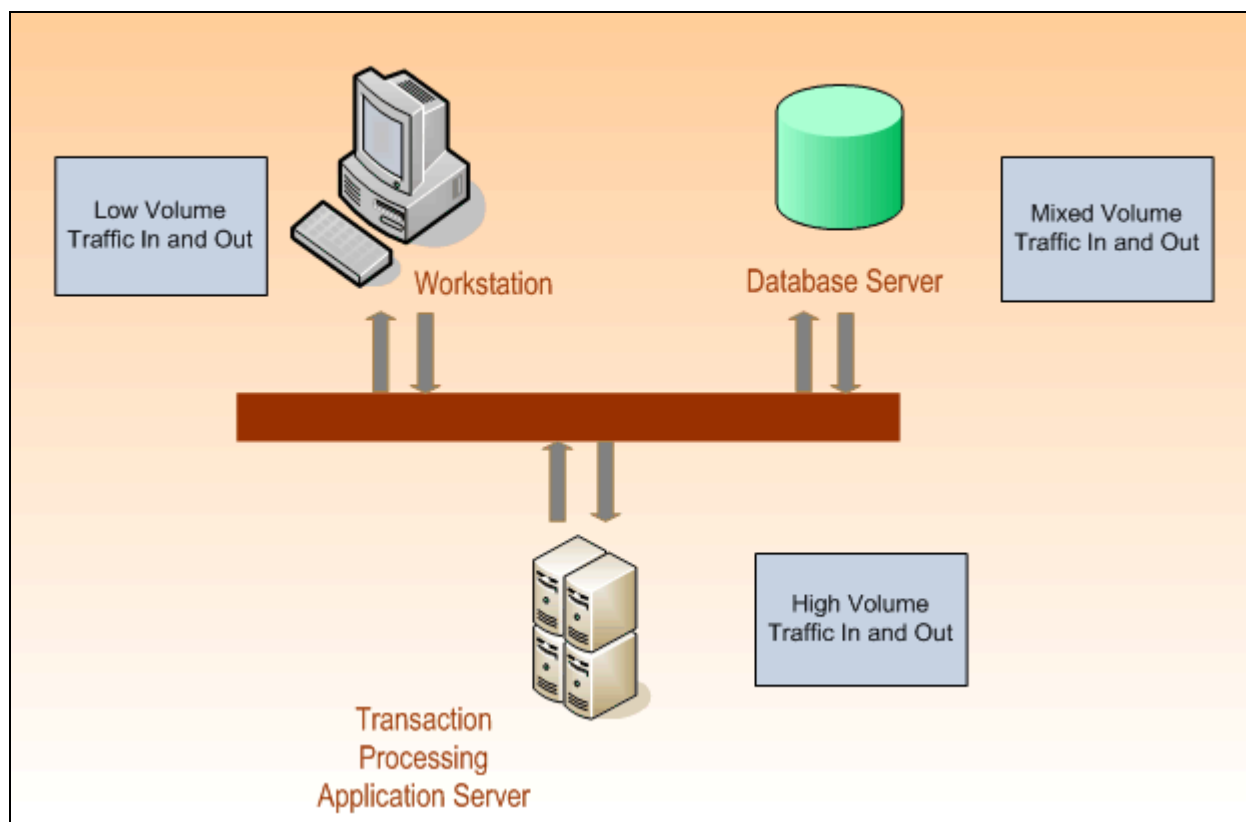
Like signature-based approaches, statistical pattern recognition is based on examining a stream of network traffic and looking for anomalous sequences of packets. Unlike its counterpart, statistical pattern recognition is not programmed with a fixed set of patterns. Instead, statistical approaches examine network traffic to gather information about “normal” usage. By analyzing sufficiently large samples of traffic, these methods can derive statistical measures that describe acceptable patterns. This approach works best on limited data subsets without wide variance. Post-event analysis can be used to create new time rules for future monitoring.

Anomalous events are detected when traffic patterns differ sufficiently from the “normal” statistical measures (typically, standard deviation). For example, if more than the normal number of SYN packets is detected, a DoS might be underway. Simple attack patterns could also be described with production rules or attack signatures, but statistical pattern recognition systems they do not need to be explicitly defined. A key advantage to this approach is that new types of attacks can be detected without adding new rules or signatures.

As this approach is based on describing “normal” traffic patterns, it functions best when the range of what is considered normal is reasonably limited. For example, if a SIM system depended on a single statistical pattern to describe traffic across an entire organization’s network, it would be difficult to identify “abnormal” behavior. Consider a typical business environment:

- Patterns of activity vary by segment—the Internet, DMZ, and various network segments will exhibit varying patterns
- Patterns also vary by protocol (IP, IPX, AppleTalk, and so on) and by the services in use
- Desktop clients have completely different patterns, with intermittent traffic to network file servers, mail servers, and other resources

Effective statistical pattern recognition requires some degree of specialization. SIM systems that support descriptive parameters about resources on a network (for example, server vs. desktop client) can more precisely describe “normal” traffic and therefore better judge truly unusual events (see Figure 4.4).



**Figure 4.4:** Statistical patterns are more descriptive when applied to particular types of services on a network.

Correlating events requires information about network traffic across the organization. Sensors, servers, and clients gather the raw detail that must be normalized and correlated in several different ways. Knowledge bases then interpret the correlated information and determine whether a security alert is warranted. No single knowledge representation scheme—that is, production rules, attack signatures, or statistical patterns—can render highly accurate and comprehensive alerts in all cases. Combining multiple techniques provides an opportunity to leverage the strengths of each. In addition to accurately identifying threats, SIM systems must process large volumes of data rapidly.

### Meeting Performance Demands

During an attack, point systems will generate increased numbers of alerts. Correlating these alerts, as we have seen, is a complex process of evaluating rules, planning and using statistical analysis effectively, comparing attack signatures, and normalizing raw data.

#### **Databases vs. In-Memory Operations**

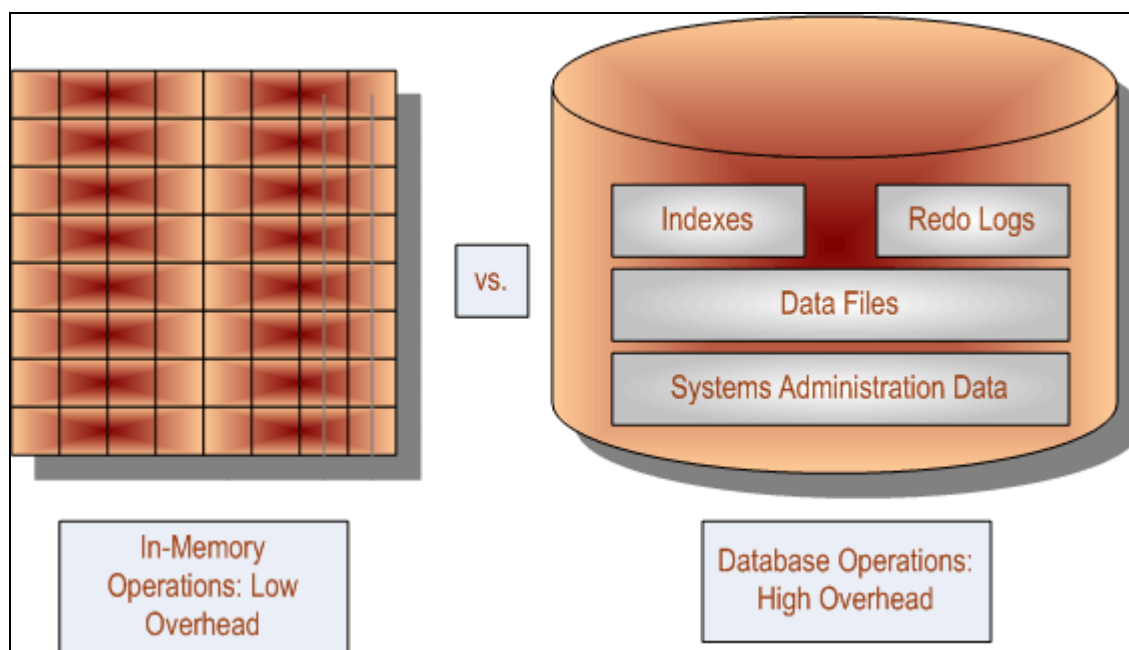
SIM systems use several features provided by the databases they employ:

- Reliable, persistent storage of knowledge base information
- Robust query language for reporting
- Consistent views of data that can be read by one user while another user updates the data

The major drawback of database systems when correlating data is speed. Databases are designed for high availability and reliability. Many application designers need those characteristics more than raw speed. SIM systems need to use databases judiciously as well as differently from other typical enterprise solutions. (In the case of forensic analysis, reliability is vital and databases can play a key role there. The following discussion is primarily focused on the role of databases in correlating data in real time).

To meet the reliability needs of forensic analysis, database designers have developed techniques that allow databases to recover from failed transactions and undo changes that have been made. For example, if a bank customer transfers \$100 from a savings account to a checking account, and the database server fails after deducting the \$100 from the savings account but before crediting the funds to the checking account, the customer could lose \$100. Databases (at least relational databases) keep extra information about transactions in something called *redo logs* that allow them to undo incomplete transactions such as this. Although these logs are essential to most business transactions, the overhead of the logs and other relational database services is not worth the impact on performance.

Correlating data in real time is generally done in memory. Accessing and updating RAM is orders of magnitude faster than the equivalent operations on disks. In addition, in-memory correlation does not require disk-based reads from indexes or updates to redo logs and other systems administration files (see Figure 4.5). Post event statistical analysis, however, does typically involve computationally intensive work with databases.



**Figure 4.5:** In-memory operations are significantly faster than database operations.

The key advantage of in-memory correlations, the lack of overhead, is also its weakness. In-memory operations are volatile. If the SIM server goes down for any reason, the contents of memory are lost. The state of statistical analysis, attack signature matching, or production rule runs are lost and must be recomputed. Furthermore, the raw data collected from point systems must be gathered, normalized, and correlated again. In essence, the analysis process starts all over again.

In most situations, the need for real-time analysis outweighs the risks of losing correlation information that can be re-created. Once correlation has been completed and findings have been made, the need to preserve information becomes more pronounced.

### Ensuring Data Integrity

The first three challenges to integration—normalizing data, correlating events, and meeting performance demands—must be met in real time. The final challenge addresses how the gathered information is preserved for future use. There are several reasons to preserve security event information even after a breach has been resolved. These include:

- Auditing for regulatory compliance
- Performing post-event investigation and forensics
- Revising policies and procedures
- Pursuing criminal investigations
- Training and education

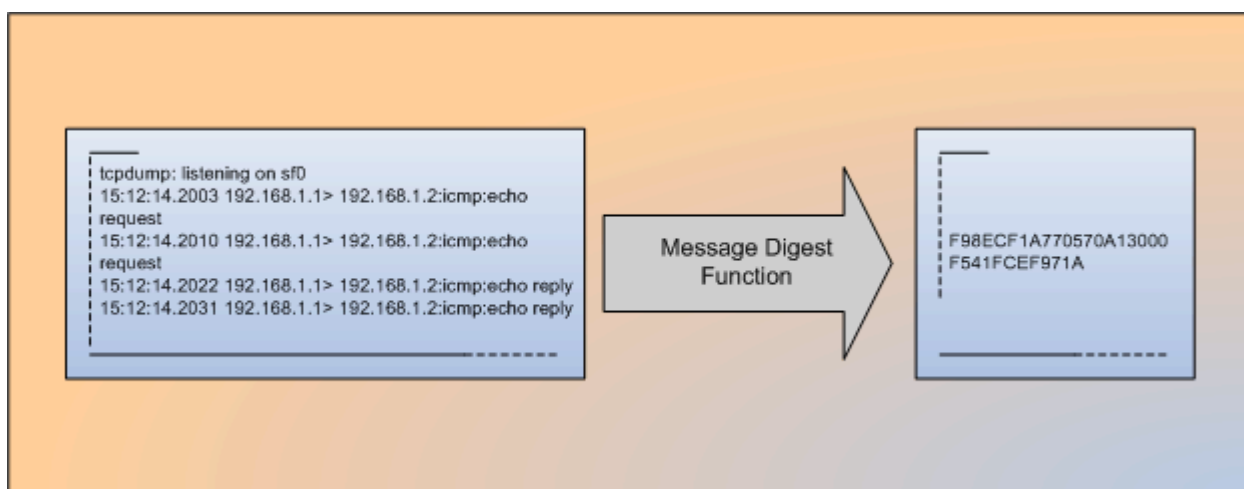
Preserving data integrity entails:

- Storing complete information about an attack, including raw data, normalized data, correlated event information, and derived facts.
- Ensuring that the data is not tampered with after the fact

Database systems combined with message digest techniques meet these requirements. However, when aggregation and reduction techniques are used, some detail is lost. Characteristics of unsummarized data may not occur in the aggregated and reduced form.

As already noted, database systems are designed to reliably store large volumes of data and prevent logically inconsistent changes to data. These systems provide a robust query language for storing, filtering, and managing SIM information.

Message digests, or similar techniques, are used to ensure that data is not tampered with. A message digest is a function that takes a block of data as input, for example a group of network packets, and computes a value called a message digest (see Figure 4.6).



**Figure 4.6:** Message digest functions calculate unique strings on input text, making it a simple matter to detect whether a file has been tampered with after the digest was computed.

The functions which compute message digests are designed to compute different values if any change is made to the input. For example, consider the following two timestamps and their corresponding message digests:

Timestamp: 12:00:00 and Message Digest: BE538A2EB38C182CDEE14975807CA554

Timestamp: 12:00:01 and Message Digest: DB9A4A8AA210B2024C19D3A923113EE6

Message digests are stored in addition to log files and other security event data. When the information is used, a new message digest is computed and compared with the stored value. If the two digests are not the same, the security data has changed. Of course, the message digests themselves must be stored securely so that someone tampering with the security event data cannot overwrite the corresponding message digest along with the original data. Computing these functions is also computationally intensive and may impact real-time performance. With reliable long-term storage and methods to maintain data integrity, SIM systems are able to meet the needs of post-event investigation and forensic analysis.

## Post-Event Investigations and Forensic Analysis

Forensics is the process of gathering and analyzing information about a security event after the breach has occurred. By its nature, forensic analysis is focused on looking for traces of information left after an attack. This information is used for several purposes:

- Identifying and mitigating systemic vulnerabilities
- Discovering patterns and significant details that are not evident from the event stream
- Pursuing criminal investigation
- Analyzing trends and patterns in security threats

These purposes have both overlapping and distinct requirements.

### ***Mitigating Systemic Vulnerabilities***

A systemic vulnerability is one that exists at multiple points in a network. These include:

- Unpatched software
- Protocol vulnerabilities
- Vague or inconsistent security policies
- Inconsistently applied procedures


### **Identifying Unpatched Systems**

Unpatched software is best managed as part of a vulnerability-management process (see Chapter 3 for more information about vulnerability management). Post-event analysis can help to identify a vulnerability that was exploited and determine the appropriate level of patching that is required to control the vulnerability. It also helps *grade* patch and vulnerability management processes as well as develop suggested improvements for the future.

### **Protocol Vulnerabilities**

Protocol vulnerabilities are those inherent in the design of a protocol. TCP, for example, relies on a handshake sequences to establish and confirm a session between two network nodes. This handshake protocol is exploited in DoS attacks that floods the victim machine with SYN packets. The victim, following the handshake protocol, can consume all of its resources by responding to false SYN packets and effectively shutdown the availability of other services due to resource constraints.

Another example is the Simple Mail Transport Protocol (SMTP). This protocol by itself does not authenticate senders, so messages can be spoofed with false sender information. This vulnerability is not a problem with a particular implementation of SMTP but with SMTP itself. Post-event analysis can help identify methods of attack that use protocol vulnerabilities that are not generally known. By analyzing details of an attack, security professionals can gain a better understanding of weaknesses inherent in network protocols.

 Understanding protocol vulnerabilities is just one example of what can be learned by analyzing attacks and hacker behavior. A promising area of security research is known as “honey pot” research. Honey pots are servers set up solely to monitor reconnaissance and attack methods of hackers. The goal of honey pot research is to understand the technical tools used by hackers and the organizational relationships between hacking groups. For more information about this topic, see Fredric Raynal, et. al. “Honey Pot Forensics Part 1: Analyzing the Network” at <http://csdl2.computer.org/dl/mags/sp/2004/04/j4072.pdf> and the Honeynet Project Web site at <http://www.honeynet.org/>.

### **Vague Security Policies**

Information gathered from forensic analysis can also shed light on vague, ambiguous, or incomplete security policies. For example, an access control policy might state that an employee’s user IDs are revoked on the person’s last day of employment. Does that same policy apply to contractors? What about contractors who finish one project and are expected to return for the next phase in several months? Hackers could compromise the contractor’s account by using social engineering techniques. Should the contractor’s account have been revoked? Yes, from a security perspective, but not necessarily according to the policy.

### **Inconsistently Applied Procedures**

Without automated tools, it is easy to fall into problems that arise from inconsistently applying security procedures. Even if your policies are well defined, clear, and unambiguous, IT staff may not have the time to attend to all SOPs.

Consider when an OS vendor announces a security patch for a widely used OS. Critical production servers may be patched immediately and others scheduled for a patch during routine maintenance. It is easy to imagine that one or two of the less-critical servers might fall through the cracks and remain unpatched during routine maintenance. These servers, in turn, can be used as jumping off points to attack critical servers.

There is also the problem of unapproved devices put on the network to fix an immediate problem. How many unpatched database servers have been installed by developers needing to test design concepts without wanting to bother a DBA? How often are wireless access points installed “just for a few days” while a team works in a conference room? Post-event analysis can help identify weaknesses in procedures and educate those who do not always appreciate the importance of following security measures.

### ***Pursuing Criminal Investigations***

For regulatory compliance, for privacy protection and disclosure, and when a security breach warrants a report to law enforcement agencies, there is a need to preserve evidence. Security events that can warrant criminal investigations include:

- Destruction of data
- Theft or exposure of private and proprietary information
- Fraud
- Identity theft


 Chapter 8 will explore regulatory and compliance topics in detail.

These investigations depend on digital evidence gathered from individual machines (host-based forensics) and from network traffic (network-based forensics). Digital evidence includes system log files, emails, instant message logs, recovered files, and network analysis.

For host-based forensics, security specialists preserve data by cloning or creating images of disks, recovering deleted files, decrypting encrypted files, and retrieving the contents of swap files and other temporary data stores. In the case of network forensics, a crucial task is preserving the stream of network traffic to allow investigators to re-create or “replay” the attack. Analysts will also need to correlate events across the network and augment raw data with other information, including:

- Adding investigation notes
- Documenting monitoring and reporting procedures
- Visualizing key relationships, such as sessions between servers, including the ports used

Ideally, a SIM system supports the gathering, preservation, and the analysis of forensic information of all types based on the sources of data.

 For more information about forensics and digital evidence, see the Web sites for the Scientific Working Group on Digital Evidence at <http://ncfs.org/swgde/otherdocs.html> and the International Organization on Criminal Evidence at <http://www.ioce.org/>.

### ***Analyzing Trends and Patterns***

SIM systems can potentially gather large amounts of information, even after aggregating and analyzing raw data from point systems. Rather than perpetuate the problem of information overload for security professionals, albeit in a different format, SIM systems support the analysis of trends and patterns across time.

## Baseline Measures

Information collected on a normally operating network can provide a baseline for future comparison with respect to key indicators such as:

- Number and rates of SYN packets per server
- Number and rates of rejected packets at firewalls
- Number and rates of port scans per server
- Number and rates of successful or failed access attempts per day, per hour, and by type (physical, host, or network)
- Number and rates of viruses detected in email on gateway systems and on hosts
- Number and rates of Web pages (or any service such as SMTP, POP, Telnet, and so on) blocked by content filtering systems (by URL and by IP address of source or destination)

Understanding the statistics and standard deviations that are significant of these indicators can help determine where resources should be applied. Do increasing number of port scans warrant changes to firewall configurations? Are content filtering systems detecting unapproved protocols tunneling through HTTP, warranting changes to intrusion prevention rules? In addition to low-level measures directed at measuring basic security posture, organizations should develop reports that measure higher-level business drivers.

## Reporting Based on Business Drivers

Security reporting should be based on relevant business drivers. There are far too many reports that can be generated to reasonably expect an organization to use them all. Depending on organizational need, some key drivers include:


- Preserving customer privacy
- Ensuring the integrity of financial systems and transactions
- Protecting the disclosure of proprietary information
- Maintaining compliance with regulations
- Enforcing policies and procedures

## Effective Communication and Security Planning with SIM Reports

Much of this chapter has been dedicated to the use of SIM systems as tools for IT operations. In practice, SIM reporting serves a wider audience including CEOs, HR professionals, auditors, and others with management responsibilities. These individuals need information that affects strategic objectives maintaining customer privacy and remaining in compliance with regulations. Useful reports include:

- Demonstrating compliance
- Identifying weakness in information security
- Visualizing complex relationships

SIM reports play a role in compliance by demonstrating that operations are monitored, security events are detected, mitigating procedures are applied, and the effects are measured. These reports can also help target weak spots in IT operations and user training. For example, if monitored traffic is showing an increase in outbound emails containing possibly proprietary information, HR can develop targeted training to address that specific problem. Visualization of complex networks and processes can also aid in the process of explaining security concepts to C-level executives and managers who are not familiar with the intricacies of information security.

 Visualization in support of security information is an active area of research. For more information about this topic, see “Visualization in Detection of Intrusion and Misuse in Large Scale Networks” at <http://csdl2.computer.org/dl/proceedings/iv/2000/0743/00/07430294.pdf> and “Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs” at <http://csdl2.computer.org/dl/proceedings/iv/2002/1656/00/16560570.pdf>. For more information about visualization for business decision making, including visualizing trend analysis, see “Management Through Vision: A Case Study Towards Requirements for BizViz” at <http://csdl2.computer.org/dl/proceedings/iv/2000/0743/00/07430387.pdf>.

### SIM Scenario

Gamma Publishing Services is a financial information services provider to Global 2000 companies. The firm provides detailed and aggregate information about stocks, bonds, corporate financials, and related business and economic trends. Gamma Publishing Services provides its clients with direct access to data on servers within Gamma’s DMZ. Point systems—including intrusion prevention systems, firewalls, network sensors, content filtering systems, and antivirus software—are deployed throughout the network.

Network analysts were alerted to a spike in security alerts; approximately 100,000 new alerts were generated per hour. Using correlated data from their SIM system, analysts determined that the 100,000 alerts were the result of 100 infections by the Nimda virus. The analysts then examined detailed reports, sorted by server priority, to determine that 5 of the 100 infections were on critical servers. The next step was to develop a short-term mitigation plan.

First, analysts reviewed vulnerability management information for each of the critical servers. It turned out that three of the five servers had already been patched for the worm. They could safely ignore those three false-positive alarms. They turned their attention to the two truly infected servers.

Emergency response procedures dictated that the network manager should be called and the patch should be immediately applied to the infected servers. If the problem was not resolved in 1 hour, the CIO was to be paged. One analyst continued to monitor network alerts, watching for infections on other critical servers. Meanwhile, the infected servers were taken offline, patched, and restarted.

After successfully dealing with the immediate containment and mitigation problem, the analysts turned their attention to post-event analysis. Using the SIM system, they generated several reports for review with the CIO and auditors. Key questions they needed to answer were:

- How long were the production servers down?
- Is the response time better, worse, or equivalent to average response times?
- Is the company remaining in compliance with all regulations?
- How can the company improve its policies, procedures, and training to prevent similar incidents in the future?

Using the wealth of data collected by SIM systems, Gamma Publishing Services was able to identify a security event, assess its impact, contain and mitigate the damage, and learn from the incident.

## Summary

SIM entails functions ranging from real-time monitoring to executive reporting. The need for SIM is a reflection of the nature of information security. That is, security events can occur with little warning and can impact an entire organization. Management structures must be in place to detect and respond to those events and to support continuously improving security procedures in both technical and organizational dimensions. The data collected by SIM systems provide an opportunity to coordinate security efforts and to maximize the efficiency of security resources to mitigate loss, respond more effectively, demonstrate compliance, uncover new issues, and manage security in accordance with real business drivers.

**[Editor's Note:** This content was excerpted from the free eBook *The Definitive Guide to Security Management* (Realtimepublishers.com) written by Dan Sullivan and available at <http://www3.ca.com/ebook/default.aspx?sacid=60453>.]