Concentric Supervision of Security Applications: A New Security Management Paradigm

Philip C. Hyland
TASC, Inc., Ph.D. Candidate, George Mason University
4801 Stonecroft Blvd., Chantilly, VA 20151
Phone: (703) 633-8300 (W), Fax: (703) 449-1080
E-Mail: pchyland@tasc.com, http://mason.gmu.edu/~phyland

Dr. Ravi Sandhu
Professor, Department of Information and Software Systems Engineering
George Mason University, Fairfax, VA 22030-4444
Phone: (703) 993-1659 (W), Fax: (703) 993-1638
E-Mail: sandhu@gmu.edu, http://www.list.gmu.edu/~sandhu

Abstract

This paper questions the status quo regarding Security Management (SM) tools that function in an isolated, monolithic fashion. People work best by interacting with others and with their systems to see the "big picture" to interpret individual events. Our view of SM called Concentric Supervision of Security Applications (CSSA) is a continuous cycle of information flow. CSSA processing of status information and control of security features does not replace existing notions. It serves to enhance the existing ad hoc and segmented "engineered" solutions so that SM systems support "the way people work".

We divide management functions into three phases, administration, operations, and assessment. Different skills, authority, and data are needed to perform tasks in each phase, but some information must flow for efficient and effective functionality. We give suggestions on some linkages by describing typical SM scenarios and how they might function. Parallels are drawn with related issues in network management systems and relationships to current management approaches are discussed.

1. Introduction

Business managers today are beginning to recognize information system security as a major corporate requirement. Significant research and development effort has produced a variety of security technologies. However, security is not strictly a technology problem; system developers and operators need complete lifecycle solutions that permit reliable implementation of a single, consistent *security policy*. A critical part of a complete security solution — along with security policies and mechanisms — includes *assurance* methods to ensure that security designs and implementations continuously meet the objectives of the security policy.

Assurance spans activities from prevention of design errors and configuration errors during deployment to detection of anomalies during operations and rational recovery from the effects of a security event after it has occurred. A complete security assurance plan would address all phases; however, we confine our discussion here to Security Management (SM) activities after system deployment. We define SM to include the activities to configure, monitor, and control the security services that a system provides. We also shall consider how the administration, operations, and assessment phases of SM can be viewed as duals of similar functions in the network management community. The great strides in recent years by network managers to actively plan, monitor and control suggests there is a considerable potential for equivalent systemic improvements for management of system and network security.

Current security tools, for the most part, take a very narrow view of SM. Most focus on configuring or controlling particular security mechanisms such as single sign-on, Intrusion Detection Systems (IDS) or key management. Security management must ensure that organizational security responsibilities match up with resources and rules to implement the defined security policy while not inhibiting the normal way people work. The framework we propose, called Concentric Supervision of Security Applications (CSSA), is an end-to-end SM concept that leverages related functions and data in the *administration*, *operations*, and *assessment* phases of management. However, we first consider what components make up the management of security and how they relate to each other.

2. SM Notions—New and Old

2.1. Assessing Administrative and Operational SM

As technology becomes more complex, we must refine terminology and re-partition problems to understand the issues better and focus attention on solvable pieces. In 1987, the OSI Security Architecture (ISO 7498-2) [**1**] identified three categories of SM — system security management, security service management, and security mechanism management — plus security of management itself. System SM is concerned with overall policy management, interaction with other OSI and SM functions and with general security functions such as event handling, auditing and recovery. Security service management deals with interactions with particular security services such as negotiation of protection levels and mechanisms. Security mechanism management includes management of keys, encipherment, digital signatures, access controls, integrity, authentication, traffic padding, routing control and notarization. Security of management (the main area SNMPv3 [**2**] deals with) defines methods and structures to ensure vital status and control functions are protected.

Although the OSI functional breakdown is useful as a reference model, it does not recognize the typical division of duties between system administrators, security managers, and systems operators. In contrast, we postulate three major phases of SM that distinguish the desirable characteristics of a management framework, namely administration, operations and assessment of security. In CSSA, we further consider SM from the view of *who* is involved, the *purpose* of their activities, and the *types of information* affected. Our intent is to indicate that different assumptions, rules-of-thumb and tools are applicable in each phase, thus justifying our subdivision of the discipline.

As a starting point, **Figure One** shows one view of the essential SM components. All systems function under a security policy (even if it is null). From the security policy, a security administration process defines configuration *rules* that specify security subsystem behavior. This process supports operational SM (represented by the traffic light labeled "Control Entities") which controls one or more security mechanisms. These mechanisms provide on-line security services that actually affect security clients. The feedback path from status sensors to the control entity provides operational health and anomaly detection information. This feedback is a resource that enables the value-added tools of security assessment.

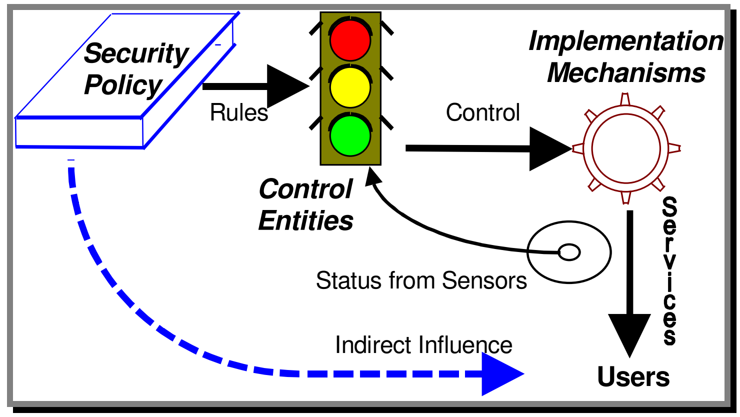

Figure 1. Security Management Components

The assessment phase is not visible in **Figure One** because it has largely been an off-line, human process due to the time and processing demands. The important concept is to use dynamic feedback to modify the security posture as events occur and the perceived security threats change.

In fact, there are two major shortfalls with this initial view of SM. First, nothing in **Figure One** integrates the management of security mechanisms together into a unified, coordinated capability. In other words, each mechanism might have its own parallel rules, configuration process and control structure that does not interact or benefit from others. Second, there is no lifecycle view of the SM process that ensures configuration data, operational data and historical audit/performance data can all be accessed and assessed to support continuous improvement. The security policy should be the starting point of a spiral of dynamic rules that adjust and improve the security posture based on current conditions, the time of day or week, and the perceived vulnerability. We believe a common SM framework will help resolve these data management issues and thereby bring more effective and mutually supporting management to security devices and applications.

The objective of assurance is effectiveness. SM tries to ensure that the security services delivered are adequate and compliant with the organization's security policy while minimizing the administrative overhead. As **Figure Two** depicts, CSSA consists of three phases with data in the middle. It operates in a repeated cycle, always seeking a secure management state.

Administrative SM can include pre-operational system development and testing, but we focus here on the *user-driven* configuration activities of system installation, setup, and maintenance. Administration minimizes system vulnerabilities by opening system access enough to meet users' needs while purposely constraining certain activities to satisfy the applicable security policy. Administrative activities are often done in a batch mode since immediate response time is not expected. Adminis-

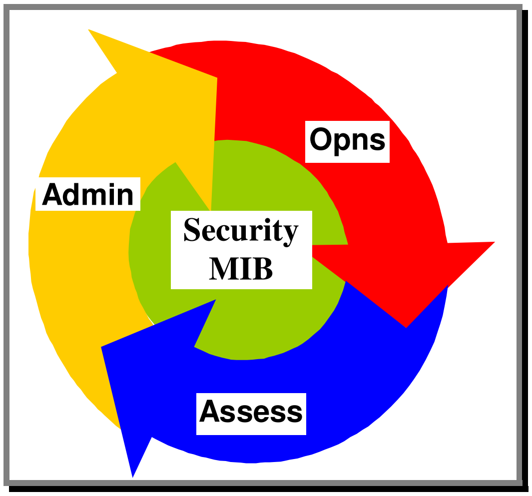

Figure 2. Concentric Supervision Cycle

trative security actions may be stopped or interrupted for a period without degrading security. In other words, administrative SM is relatively routine and part of the normal day-to-day functioning of the enterprise.

Differences between administrative SM and operational SM are evident especially in their purposes. Operational SM is a more active, *event-driven* component of assurance that is concerned with detection and reaction to current conditions applicable to security mechanisms. Operational SM consists of real-time interactions between the security *service providers* (mechanisms), *status sensors* and *control entities.*

Tools should be designed to monitor and maintain security posture at levels defined by the established security policy. As the desired protection level increases, the corresponding management functions must become more stringent (see **Table One**). For example, at level C, access control defined by Ad3 would be more restrictive than for level B and below. Likewise, operational thresholds may be tighter and assessment responses more conservative. Operational SM exists to detect non-compliance with security policy; thus, operational management tools must be sensitive and continuously available to reliably track and respond to unpredictable security events.

The assessment phase of SM is *performance-driven*; that is, it is concerned with measuring whether objectives are met and how potential changes may affect the security system. Evaluation of audit trails and pattern matching are assessment activities. Such assessments may have short-term or long-term scope. Short-term, quick response assessments generally support reaction to imminent threats detected during the operations phase. Based on other events correlated in time, space, or modus operandi, an appropriate response is initiated or recommended to an operator. Long-term assessment may support security policy planning, trend analysis of threats, or quality of protection issues. A simulation tool may test a proposed administration state against a range of operational conditions to assess its desirability.

2.2. Existing Management Architectures and Related Work

The foregoing concepts do not detract from the work that has been done in many areas of the security management challenge. IDS research has been a major thrust while configuration-checking tools such as COPS, and SATAN have improved the security position of many sites. Several major commercial products such as CA Unicenter TNG, Sun Security Manager, and Tivoli TME10 do support SM functions within their management architectures, but none view management as a continuous cycle of activities as we have suggested. The OSI Security Architecture has little sense of temporal issues or sharing data between administration, operations, and assessment functions. Other more generic management models such as SNMPv3, the DCE security management framework [**3**], and two European research projects (SAMSON [**4**] and SMCN [**5**]) focus mostly on protocols and operational tools for a few security mechanisms. Notably, the "Security Mechanisms for Computer Networks" project (1985-1990) included security management in its Comprehensive Integrated Security System (CISS) design. CISS emphasizes use of existing mechanisms and APIs for common security functions. Of the ten functional agents identified in CISS, two are devoted to security administration functions, two provide general support to all components, and six support operational SM.

3. Elements of CSSA

Table 1. Security Protection Levels

Security Level	Administration	Opns	Assessment
A	Ad1	O1	A1
B	Ad2	O2	A2
C	Ad3	O3	A3

We organize management functions into three phases because similar data, processing, and skills are used in each phase. In this section, we present details of how CSSA phases mutually support each other.

3.1. Administration

The administration phase includes both initial configuration of security services and routine updates to add, delete, or modify user and resource information. The authority for such actions must be carefully controlled and audited. Before updates are applied, crosschecks for compliance with security policy should occur. This implies a security policy that has been tuned to a set of guiding principles. If the security policy is defined as a set of management rules, this process may be automated. Otherwise, a manual checklist procedure can verify consistency of changes.

Some configuration parameters apply widely (e.g. current security level) while others are relevant only to specific security mechanisms (e.g. key expiration date). In many cases, important data for user authentication, key distribution, access control, security filters, and directory services will be stored in one or more databases. The CISS model calls the aggregation of all security management data the Security Management Information Base (SMIB). **Figure Three** shows the types of SMIB access during each management phase. Administration involves mostly writing of data, while operations phase is equally read-write and assessment is mostly read-only. Some assessment results may be passed to administration for action or saved for future use.

A common means to access and update configuration data is important. Security mechanisms that do not implement a structured data store like the SMIB and a secure means of access must use a local configuration file for vital management data. This leads to inconsistency among applications and requires a proxy to interface with the control entity using standard protocols.

Suitable control over the integrity and correctness of security configuration information is vital to compliance with a security policy. Delegation of authority to make changes needs to be very fine-grained. Senior security administrators may need full authority to update configuration, operational and assessment roles and privileges while delegating local tasks, such as user account updates and modification of access lists for local resources to subordinate security managers. *Least privilege* facilitates decentralization without giving too much power to any individual.

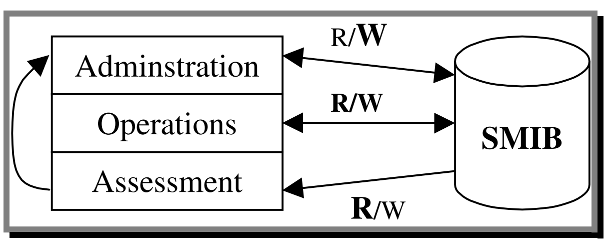

Figure 3. Management Access to Data Store

Administrative management also may be viewed as a workflow problem. A sequence of steps and crosschecks can be designed to ensure proper authority and compliance with policy rules. Although space prevents full details here, the process can be reduced to a set of Integration Definition (IDEF) diagrams. Relationships become complex if the security policy requires special measures such as two-person authorization (i.e. for database updates) or Chinese wall access controls. Separation of duties permits independence while enforcing integrity checks for roles in which one person could have undue authority.

3.2. Operations

Operational SM is characterized by real-time interaction with security components that provide security services to end-users. Management tools for security operations need several capabilities such as real-time interactive control, detection and alarm mechanisms, event logging and correlation, and graphical display systems. In addition, limited access to configuration and historical data would make trouble-shooting and problem prevention easier. For example, a system operator should have read access to contact information for any user within their domain and should have write access to certain configuration data within their responsibilities. Consequently, the operator would be able to notify the user of an alert from a certification authority that their certificate is near expiration.

The driving force in design of operational SM is dealing with active or suspected security threats. Like the performance and fault management functions within the field of network management, operational SM has a strong temporal factor. Monitoring the real-time status of performance and component health, along with tracking of security events is necessary to permit active responses to changing conditions. For example, in a benign security environment, an indication from a network component sensor of a performance bottleneck might result in the control element setting a more permissive filtering stance (i.e. reduced logging). This would improve performance until the backlog is reduced. On the other hand, exceeding a security monitoring threshold (such as number of repetitive logon failures) that might signal an attack may justify increased event logging, alarms to system operators, or automated responses to deter damage to system integrity.

3.3. Assessment

In many ways, the assessment phase is the most critical in creating an effective management system. It is the essence of intrusion detection research. It is the chief feedback loop to modify system attributes based on current conditions. Events that are detected and recorded during operations need to be matched against known patterns, related targets, or even similar times of occurrence. Depending on the danger and uncertainty of an event, immediate or long-term configuration changes may be effected. Context is significant. Certain threats may trigger pre-determined responses such as session termination, increased monitoring, or deception mechanisms. Unfamiliar or uncertain events should result in more cautious and conservative actions. Artificial intelligence methods are helpful in correlating similar situations. A major benefit of a centralized management approach is the capability to evaluate individual events against a broader system view and more sophisticated software on higher performance systems than possible with many dispersed nodes. On the other hand, an assessment approach with a distributed correlation database may provide needed redundancy and scalability.

Assessment is not necessarily a static process. As was indicated in **Table One**, the security level may drive a more aggressive stance as perceived vulnerability grows (i.e. before a new product release or during sensitive negotiations). In this way, the SM system becomes more proactive, in contrast to its traditional reactive style.

4. SM Notions in Action

Why is a distinction between management of security administration, operations, and assessment important? Different people and priorities are involved, but sharing and evaluating common data can vastly improve overall management. To achieve the goal of an integrated SM architecture, both the administrative and operational environments must be understood and supported. Finally, the role of assessment in the context of configuring and monitoring security mechanisms to comply with an overall security policy is vital.

Now we present a few examples to highlight the difficulties of trying to integrate security management. We consider some simple, but practical applications that justify our Concentric Supervision of Security Applications (CSSA) concept. We will determine where sharing data between the configuration, operational, and assessment phases is currently possible and where it is problematic.

4.1. Virus Alert

Our first scenario begins with a security incident detected by a security mechanism during normal operations. At 3:40 PM, an automated virus checker identifies and reports a virus imbedded in an email message sent to Alice from the XMU.edu domain. As an active security mechanism, the virus checker supports the operations component of CSSA by generating a notification to the control entity as shown in **Figure Four**.

In the first stage of reaction to a notification, the event manager (dispatcher) module logs pertinent information (see **Table Two**) and starts an audible alarm at the network operator's console. The network operator, if on duty, reacts according to the severity of the alert and the importance of other on-going activities. In addition, the event manager alerts the security officer via email (or pager) to review whether current actions are appropriate and whether executive management needs to be informed.

In a cooperative environment, peer monitoring sites may be alerted about information that could represent a low-key, but broad-scale attack. Optionally, application users (email sender and recipient) may be notified. In some cases, this notification may be undesirable because

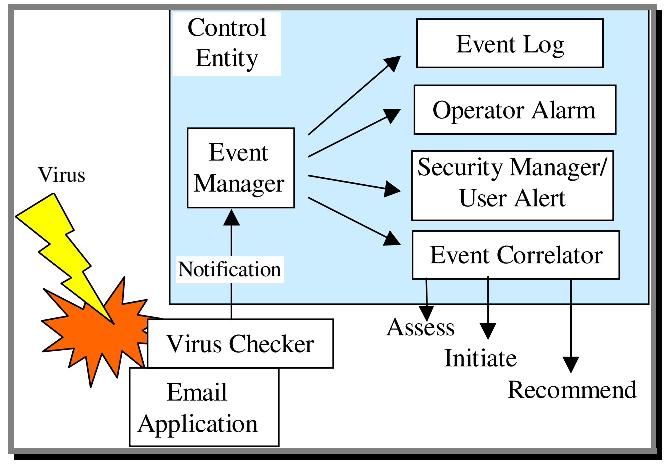

Figure 4. Response to Virus Incident

it may reveal more than desired about the response capabilities.

Lastly, but most importantly, the event manager signals the event correlator to evaluate the incident. The signal to the correlator represents transition into the next phase of CSSA cycle, security assessment. The event manager forwards data from the initial notification. The event correlator uses this data and information extracted from the event log, including alerts from peer control

Table 2. Event Notification/Logging Format

Name	Value
Source Address	IP Address of reporting entity
Type of Alarm	Security Event
Date/Time Stamp	Date Time Group (DTG)
Severity Level	(Critical, Major, Immediate, Warning, Minor, Informational)
SubEvent Type	Virus detection
Suspected Source	IP Address, Domain, USERID
Comment Field	Additional event-specific details

entities, to evaluate and recommend a course of action congruent to the desired security level.

The correlator may initiate some protective measures automatically through interim configuration updates. It leaves recommendations for weightier policy or configuration changes for human action. Automated updates may include damage mitigation that requires immediate action to prevent further damage. Examples of conservative responses might include disabling email from the suspected source address or isolating an infected disk volume. Care is needed to ensure protective measures do not cause unintentional denial of service that is more harmful than the original incident.

When the correlator recommends a particular operator action, the operator may not respond promptly or at all. Based on operator history, the security level, and the event severity, the response may be automated. Suppose another virus originated 3 months earlier from the same XMU.edu domain. In this case, changes to the security policy or baseline configuration may be appropriate. Since XMU.edu is not a vital business partner, the event correlator chooses to block all traffic from that domain. At this point, the relevant commands and information pass from the assessment phase into the administration phase of the CSSA cycle.

In the administration phase, a security administrator typically sets and updates parameters that control the operations of security mechanisms to match the security policies of the organization. In the CSSA concept, all security events can potentially result in configuration changes. An automated configuration change initiated in the assessment phase may bypass the manual process when necessary. Non-urgent changes are reviewed by the security management and implemented by the security administrator. All changes are logged as an internal security event. Since internal security events are not assessed by the event correlator, this completes the scenario. **Figure Five5** shows the sequence of processing through all phases of our model.

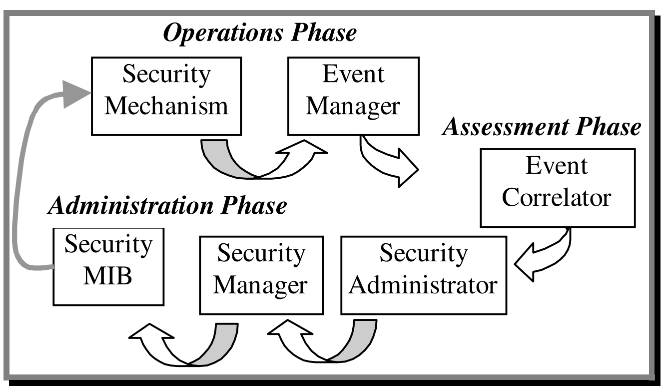

Figure 5. Event Processing Sequence

4.2. Remote Site Firewall

In scenario two, Bob is configuring the firewall component for a new remote office LAN as outlined in **Figure Six**. Proxy services need to be implemented based on the corporate security policy specified in **Box 1**.

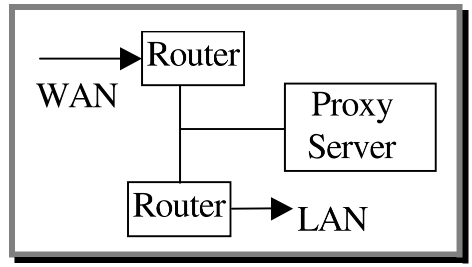

Figure 6. Remote Site Firewall

Typical Internet and Intranet services will include a local Email server, internal FTP to the corporate FTP server, outgoing HTTP and Telnet, and Domain Name Services (DNS). Because of the small size of the branch office and a desire to minimize costs, the firewall setup will use router-based packet filtering and the TIS Firewall Tool Kit (FWTK).

Design of packet filtering rules and implementation of appropriate proxy services is a daunting task, even for an experienced network engineer. Although some vendors have delivered configuration tools to simplify definition of rules, consistency across vendors is low. This leads to errors, inconsistencies, and vulnerabilities without a reliable means of verification. Existing tools do not use common conventions and all store their configuration data in different formats.

Given that an initial security configuration has been developed, the next step is to install and test it. This process opens new possibilities for errors and undiscovered faults due to limited testing rigor. While a number of

Box 1. Security Policy

1. All services not specifically required for operations will be disabled.
2. All email and ftp traffic to/from government sites will be encrypted.
3. Strong authentication is required for dial-up remote access service (RAS).
4. Incoming TELNET and FTP is prohibited except for FTP to the FTP server on the bastion host.
5. Passwords and user keys must be updated at least every six months, must be at least 8 characters and be of medium randomness.
6. System administrators must re-validate all accounts at least every six months.
7. No auto forwarding of email is permitted to external accounts.
8. All outgoing email will be checked for security releasability (dirty word check).
9. All incoming email, ftp and http traffic will be virus checked
10. No site access is permitted from non-US domains.
11. More than three security events from a single domain will require explicit authorization for continued access.

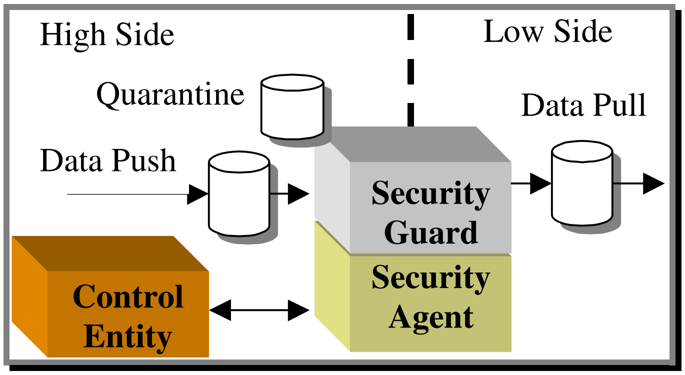

Figure 7. Security Guard

vulnerability testing tools such as COPS, Tiger, and SATAN have been useful to detect configuration problems, they are static and do not test configurations against local security policies. In addition, none uses local security events as input to dynamically tailor assessments. That would require standardized security policy formats, common event and configuration parameter definitions, and new verification tools.

Another approach for validating security configurations is simulation of the operational environment using modeling tools. A simulation environment for security management comparable to those that exist for network modeling would permit the effect of changes to be considered under a broad range of assumptions about the threats and vulnerabilities. Mechanisms could be evaluated under increasing levels of stress to determine performance and points of failure. Such a simulation environment has not been attempted yet and would require modules to simulate each category of security mechanism. Standard parameters must be created to define critical performance capabilities and measurables.

Once the validation of a newly developed security configuration is complete, approval and deployment can proceed. Audit records must capture who makes each change, when, why and how secondary approval (i.e. the security manager) was accomplished. This information becomes a permanent part of the site security log for future reference.

4.3. Security Guard

In our final scenario, a government agency is planning for management of a security device that will provide a trusted interface between systems at two different security levels. **Figure Seven** shows the main components of a security guard. Active supervision of the information security guard function will minimize risk, but minimal human interaction is also desired. Typically, a semi-automated review process may include automated checks for data integrity, template matching and filtering for unreleasable words or phases, followed with a final human review of images or graphics.

In the present case, a security guard with highly advanced releasability evaluation software will operate to permit only properly marked, formatted, and signed data items to pass from the higher security domain to the lower domain. The guard will operate independently given a proper configuration, but will generate status and alarm information to indicate its operational health and any abnormal conditions. To minimize the logic required within the security guard, the CSSA monitoring actions will be initiated by the control entity except for security event notifications.

The first task before beginning operations of the security guard is to configure required parameters. There are two types of parameters-mechanism configuration data and user data. The first type of data determines how the security device will communicate and operate. User data is used in the basic guard functions to validate releasable data. **Table Three** lists some representative data items. Configuration data that is unique to the security guard may be stored locally for immediate access. Since user data may be used by several mechanisms besides the guard, it is stored in the SMIB and accessed on demand. The list of authorized releasers and their public keys helps to document an audit trail of release actions using digital signatures.

Table 3. Configuration Parameters

Performance Data	User Data
Notification Destination(s)	Authorized Releasers
Control Entity Public Key(s)	Public Keys of Releasers
Agent Private Key	Approved Destinations
Throughput Threshold	Custom Filter Data
Input Buffer Threshold	Template Value Data

Once configured, the security guard can be tested and placed into operations. To monitor the guard operations for maximum benefit, two measures of quality are paramount. The first priority in security applications is effectiveness of the solution. Therefore, the number of false negatives (releases allowed that should not have been) should be nearly zero (fewer than a human would allow). This prevents catastrophic failures that negate all potential benefits of the system. The number of false positives (releases prevented that were okay) should be low also. This factor determines how efficient the system is. If too many releases are stopped when subsequent manual review shows no problem, it indicates inadequate intelligence in evaluation algorithms. Manual review results in lower efficiency and slower payback. Once accredited as effective and certified as efficient, system operations can begin.

The configuration data defined above is used during operations to guide how the security guard functions. If a buffer for data nears capacity, an alert may be generated to the control entity to slow input. If the throughput of the guard and the security threat are low, then less restrictive processing may be set. For example, if several files are backlogged while waiting processing, time-consuming steps may be skipped to speed things up.

Other events may signal the control entity of problems like quarantined files that fail security screening and requiring human intervention. Reasons for flagging files and the results of manual evaluations can be logged and translated into revised configuration data that will permit better processing in the future. Even if tools to implement the cyclic processing of security management data are not highly automated, the process of continuous feedback and rule enhancement should improve responsiveness and quality of operations over time. The critical factor for enhancing system performance is to use assessment results to update configuration parameters or relevant security policies. While the feedback process may occur manually, synergies can grow as automation permits better linkages between CSSA phases and security applications.

4.4. Network Management Parallels

Since SM is considered a component of the network management framework, there are bound to be some similarities between them. The increasing number of security devices in need of integrated management tools can be compared to the state of network management several years ago. SM implementation is fragmented and proprietary, much like network management was prior to emergence of the standards-based SNMP and CMIP [**6**] communities.

Since then, network management has advanced hugely within three development areas. SNMP and CMIP still deal mainly in operational network management, along with some aspects of network administration (i.e. accounting and configuration). Although fault management is a big part of network management, standards do not define administrative aspects such as trouble ticket systems. The third part of network management is network engineering and performance assessment. Recent efforts have enabled use of operational network performance data as input to design analysis and performance modeling tools. In a way, this provides a feedback path like the one we suggest for SM whereby operational data can influence the initial configuration. From these trends in network management, we can argue that SM needs to move toward standards to spur more development of integrated tools for administration, operations, and assessment/planning.

Both security and network management involve troubleshooting to determine cause-effect relationships of events of unclear origin. Just as good configuration and asset management helps network managers during troubleshooting, operational SM stands to gain from better security administration. Security administration is a necessary investment in user and resource identification, privilege assignment, and system security configuration. In an environment of partial knowledge, both network and security management need the ability to sift through events in the past or in other parts of the network to detection patterns in audit trails and event logs. This enables full exploitation of computer-assisted SM by using feedback to update configuration data.

Security and network management processes both must allow human operators to view current status in the context of past actions and related configuration information to guide operators toward rational changes. To do so, operational SM systems may use secure components of an existing network management system to perform data collection/display and support assessment and decision support functions. Below we discuss the supporting components and functions that are necessary to deliver the features of CSSA.

5. CSSA Infrastructure

The CSSA concept requires a number of support capabilities, some of which exist and some which do not. **Figure Eight** is an overview of the required components.

5.1. Components

Administration functions are located in the upper left corner of **Figure Eight.** This is where the administration manager adjudicates access to the security policy, authentication data and access rights, etc. This data may be stored centrally in a SMIB or secure directory, at distributed security mechanisms, or both. Existing network management standards do not specify MIB modules for security applications (e.g. a Firewall MIB) or how MIB data is physically stored. The SMIB may comprise access control lists and security policies, as well as active security status information. Our work indicates MIB definitions must be closely aligned with application functionality; however, a standard data access API and certain core parameters for the SMIB could simplify development of SM development by enabling use of common modules. Wide availability and open standards should be the main drivers of any choice.

The core of the operational management function is the security event manager. This module makes first-level decisions on how to handle notifications of security events and mechanism status. Next to the event manager in **Figure Eight** is the Management Protocol Interface

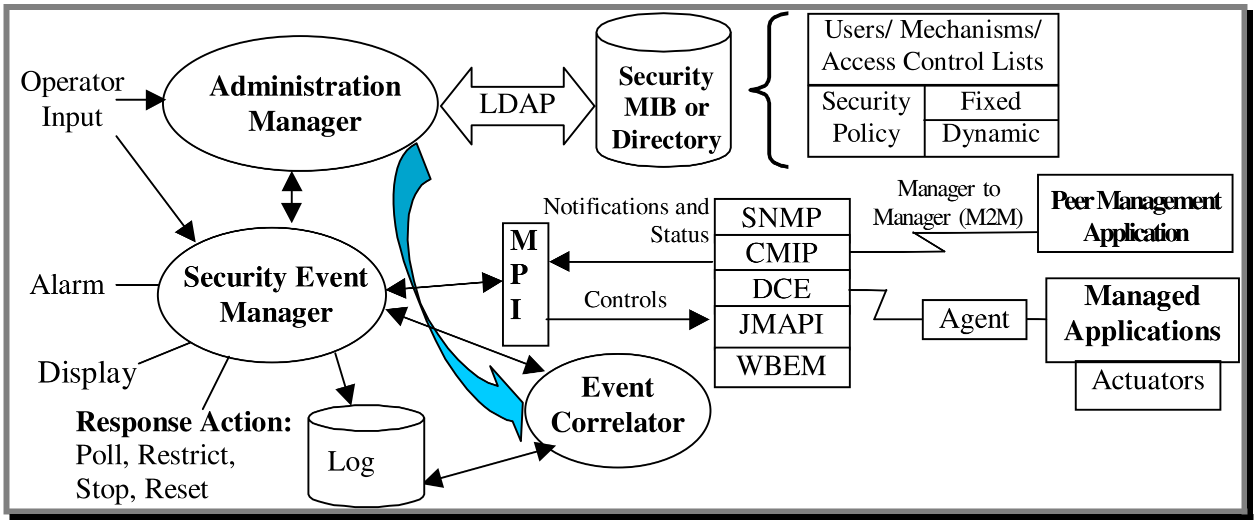

Figure 8. CSSA Infrastructure

(MPI) module. The MPI translates to/from selected management protocols that generate commands (controls) and receive notification and status information from external applications or peer management nodes. The X/Open Management Protocol (XMP) [**7**] was developed to provide similar functions for SNMP and CMIP.

Assessment functions are represented by the event correlator module. It uses current event data, SMIB data, and historical information from security logs to determine complex patterns and trends. Ongoing work on pattern-matching engines and intrusion detection methods [**8**] fit into this role nicely. The bi-directional arrow to/from the correlator implies that it generates recommendations based on its analysis of incoming events and the current security posture. Recommendations may involve policy updates, mechanism configuration changes, or access control actions.

A fundamental need for the CSSA is a means of secure transactions between many security mechanisms and one or more control entities. In **Figure Eight**, we show several candidate interface protocols (e.g. SNMPv3, CMIP, DCE etc.). More than one may be used to operate in a mixed environment. Below we describe some possible sequences.

5.2. Secure Transactions

A typical transaction sequence such as associated with the virus checker scenario would begin with the managed application signaling an event to its local agent (see **Figure Nine**). The steps listed below in three grouping are keyed to **Figure Nine**. They comprise actions to share data in support of the operations, assessment, and (re) configuration phases.

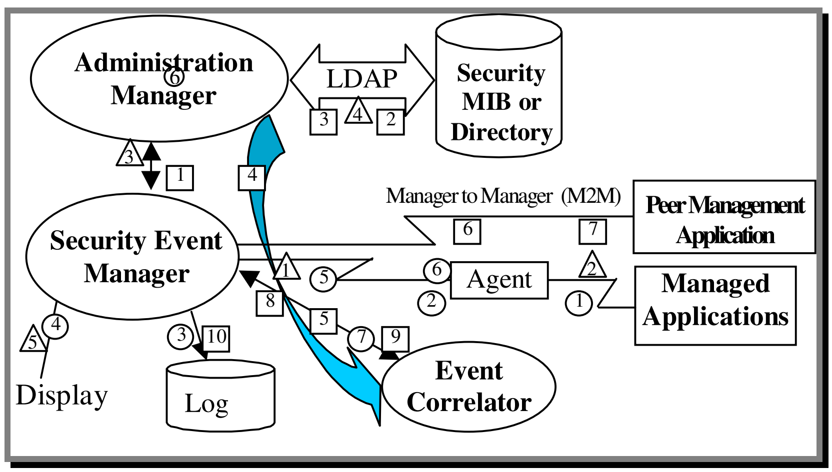

Figure 9. Secure Transactions

Operations Phase:

① **Signal**: Managed Application > Device Agent
② **SNMP Trap**: Device Agent >MPI >Event Mgr
③ **Log Record**: Event Mgr > Log
④ **Display Status Change**: Event Mgr >Display
⑤ **SNMP Poll**: Event Mgr >MPI >Device Agent
⑥ **SNMP Bulk GET**: Agent > MPI >Event Mgr
⑦ **Event Data**: Event Mgr > Event Correlator

Note that steps① through④ represent the initial alert. Steps⑤ and⑥ are actions by the event manager to gather additional details of the event that are not in the alert message. Such actions are typical of many network management systems.

Assessment Phase:

1 **Data Request**: Event Correlator > Admin Mgr
2 **LDAP Access**: Admin Mgr > SMIB
3 **LDAP Response**: SMIB > Admin Mgr
4 **Configuration Data**: Admin Mgr > Correlator
5 **Data Request**: Event Correlator > Event Mgr
6 **CMIP GET**: Event Mgr > MPI >CMIP Agent
7 **CMIP Response**: CMIP Agent >MPI >Event Mgr
8 **External Data**: Event Mgr > Correlator
9 **Assessment Result**: Correlator > Event Mgr
10 **Log Record**: Event Mgr > Log

In the assessment phase, the event correlator first initiates a request for similar data from the SMIB. Steps $\boxed{6}$ and $\boxed{7}$ represent an optional request to an external management node for related event information. This data could show evidence of a distributed, but coordinated probe or attack. The correlator passes assessment results back to the event manager for logging and possible action.

Configuration Phase:

1 **SNMP SET**: Event Mgr >MPI >Device Agent
2 **Local Change**: Device Agent > Managed Appl
3 **Configuration Update**: Event Mgr > Admin Mgr
4 **LDAP Write**: Administration Mgr > SMIB
5 **Display Status Change**: Event Mgr > Display

If configuration changes are deemed necessary, data is sent to the managed application using a SNMP SET operation and to the administration manager to update the SMIB. Not shown above are possible steps for operator or administrator interactions to control specific changes.

5.3. Areas with Good Research Potential

Several topics related to SM and CSSA need further development to attain the critical mass seen in network management. One weakness that strains system administration resources is ease of configuration and upkeep. Lack of consistency, common tools, and shared services (e.g. directory services) contribute to this problem. Standardized SMIB definitions for key applications (e.g. firewalls) would help vendors to move toward standard interfaces and interoperable tools.

The SMIB (or secure directory) repository enables structured data sharing. Common methods for storing SMIB data would enable better interoperability. Direct access to the SMIB is restricted to the trusted administration module; however, role-based access could be useful. A unified SMIB access module might use the Lightweight Directory Access Protocol (LDAP) or other distributed data archive.

A third area that may benefit from additional research is use of simulation methods to assess a potential security system for adequate protection. Virtual security systems could be tested at various threat levels and traffic conditions to determine processing loads and reaction to high stress situations. Integration with operational systems may lead to real-time feedback.

6. Conclusion

The increasing need for SM is clear as the number of security devices and applications grow. Current methods of manual access control and system security configuration are labor intensive and error-prone. To support the needs of organizational policy and operational SM, closely integrated tools for security administration and real-time operations are needed. Increasing complexity and sophisticated attacks make better assessment methods a priority.

As a contribution toward a cohesive security management framework, we proposed a three-phase CSSA lifecycle. While some pieces do exist, further work is required to refine the specific data needs in each phase, design the structure of SMIB data to be shared, and establish a standard data management API for information distributed among modules. Our proposed feedback process should lead to significantly enhanced security levels as developers adapt tools to share information between the SM phases more effectively.

7. References:

[1] International Standards Organization, Information Processing Systems – OSI Reference Model. Part 2: Security Architecture, ISO/TC 97 7498-2, 1988.
[2] RFC 2271-2275, available at ftp://ds.internic.net/rfc/rfc2272.txt.
[3] Common Security: CDSA and CSSM, ISBN 1-85912-194, The Open Group, 1997
[4] URL: http://www.darmstadt.gmd.de/TKT/security/samson/samson.html
[5] Muftic, Sean, et al, Security Architecture for Open Distributed Systems, Wiley and Sons, New York, 1993.
[6] International Standards Organization, Information Processing Systems – Common Management Information Protocol Specification, ISO/IEC 9596, 1989.
[7] Systems Management: Management Protocols API (XMP), The Open Group, ISBN 1-85912-027-X, 1994.
[8] Lunt, T., et al, A real-time Intrusion Detection Expert System (IDES) - Final Report, SRI International, Menlo Park, CA, Feb. 1992.