CHAPTER

Security Monitoring of Industrial Control Systems 12

INFORMATION IN THIS CHAPTER

- Determining What to Monitor
- Successfully Monitoring Security Zones
- Information Management
- Log Storage and Retention

The first step of information analysis requires a certain degree of data collection so that there is a healthy body of data to assess. Collecting evidence relevant to cyber security requires knowing what to monitor and how to monitor it.

Unfortunately, there is a lot of information that could be relevant to cyber security, and because there are many unknown threats and exploitations, even information that may not seem relevant today may be relevant tomorrow as new threats are discovered. Even more unfortunate is that the amount of seemingly relevant data is already overwhelming—sometimes consisting of millions or even billions of events in a single day, with even higher rates of events occurring during a period of actual cyber-attack.[1] It is therefore necessary to assess which events, assets, applications, users, and behaviors should be monitored—as well as any additional relevant systems that can be used to add context to the information collected, such as threat databases, user information, and vulnerability assessment results.

An additional challenge arises from the segregated nature of a properly secured industrial network. Deploying a single monitoring and information management system across multiple otherwise-separated zones violates the security goals of those zones and introduces potential risk. The methods used to monitor established zones must be considerate of the separation of those zones, and the data generated from this monitoring need to be managed accordingly as well. While there are benefits to fully centralized information management, the information being generated may be sensitive and may require "need to know" exposure to security analysts. Therefore, centralized monitoring and management needs to be overlaid with appropriate security controls and countermeasures, up to and including full separation—forgoing the efficiencies of central management so that the analysis, information management, and reporting of sensitive information remains local in order to maintain absolute separation of duties between, for example, a highly critical safety system and a less secure supervisory system.

In order to deal with massive volumes of log and event data that can result from monitoring established network zones, and the challenges of highly distributed and

segregated zones, best practices in information management—including short- and long-term information storage—must be followed. This is necessary in order to facilitate the threat detection process, and also as a mandate for relevant compliance requirements, such as the North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP), NRC Title 10 CFR 73.54, Chemical Facility Anti-Terrorism Standards (CFATS), and others (see Chapter 13, "Standards and Regulations").

DETERMINING WHAT TO MONITOR

The trite answer to "what to monitor" is "everything and more!" Everything that we monitor, however, results in information that must be managed. Every data point results in a log record, or perhaps a security or safety alert. Assets, users, applications, and the communication channels that interconnect them all require monitoring. Because there are so many assets, users, applications, and networks that need to be monitored, the total amount of information generated every second in even a moderately sized enterprise can be staggering.[2] While products exist to automate security event and information management, the total amount of information available can quickly overwhelm the information analysis and storage capacity of these tools. Therefore, security monitoring requires some planning and preparation in order to ensure that all necessary information is obtained, without overloading and potentially crippling the tools the information is intended to feed.

One approach is to segregate monitoring by zone. Just as the separation of functional groups into zones helps minimize risk, it also helps to minimize the total information load that is generated by that zone. In other words, there are limited assets and activities within a zone, and therefore there are less total logs and events.

To further complicate matters, operational technology (OT) activities and metrics must also be considered when securing industrial networks—representing new data types from yet another potentially overwhelming source of new assets such as remote terminal units (RTUs), programmable logic controllers (PLCs), intelligent electronic devices (IEDs), and other industrial assets; applications such as human–machine interfaces (HMIs), and Historians; and networks such as fieldbus and smart grid networks.

TIP

When considering network monitoring and information management, it is helpful to benchmark the information load currently being produced in both IT and OT networks. IT networks require identifying which devices need to be monitored. This means understanding what servers, workstations, firewalls, routers, proxies, and so on (almost every IT device is capable of producing logs of some sort) are important—the process of determining critical assets described in Chapter 2, "About Industrial Networks," and Chapter 9, "Establishing Zones and Conduits," is helpful here. Once it has been determined which devices need to be monitored, the event load generated by these devices needs to be calculated. One method is to measure the event load of a period of time that contains both normal and peak activity, and divide the total number of events by the time period (in seconds) to determine the average event per second (EPS) load of the network. Alternately, a worst-case calculation can be based entirely on peak event rates, which will result in a higher EPS target.[3]

Most assets in OT networks, mainly the embedded device types, like PLCs, RTUs, and IEDs, which make up the majority of network-attacked assets, do not produce events or logs at all, and therefore they cannot be measured. However, they do produce information. This can be easily derived by looking at historized data from the control plants, and/or through the use of specialized industrial protocol monitors. Determine which assets you wish to monitor, and use the Data Historian system to determine the amount of information collected from these assets over time. This information will need to be normalized and centralized—either automatically via an SIEM or similar product, or manually via human time and effort—so it may be prudent to limit the amount of historized data that need to be exposed for security assessment. Some Historian tags—especially system tags concerning authentication, critical alarm tags concerning point or operational changes, stopped or failed processes, and so on—are obvious choices, while others may have little relevance to security. This step is effectively a form of security event "rationalization," similar to the process performed on the process event systems of ICS to improve operational effectiveness.

Once the initial benchmark is obtained, add room for growth, and room for headroom—perhaps 10% (this will vary by situation). When sizing the IT network, it is also prudent to plan for "peak averages" where peak traffic rates occur for extended periods of time (i.e. the peak becomes the average), as this condition can occur during an extended attack, or as a result of a successful breach and subsequent infection with malware.[4] Unusual peak averages may also occur on OT systems during abnormal events, such as plant startups and shutdowns, or during system patching or on-process migrations and upgrades. OT systems may report different conditions but are less likely to report higher numbers of conditions unless the control process being historized has been significantly altered.

So what really needs to be monitored? The following guidelines help to identify what systems should be monitored.

SECURITY EVENTS

Security events are those events generated by security and infrastructure products: network- or host-based firewalls, network routers and switches, malware prevention systems, intrusion detection and prevention systems, application monitors, and so on. Ideally, any event generated by a security device should be relevant, and therefore, these devices should be used for promiscuous monitoring. Realistically, false positives can dilute the relevance of valid security events.

NOTE

The term "false positive" is often misused. False positives are often associated with what are seemingly irrelevant security data because security logs and events originate from many sources and are often generated quickly and in large quantities. When an alert is generated because a benign activity matches a detection signature of an intrusion detection system (IDS), the result is a false positive. Similarly, if an anti-virus system falsely indicates that a file is infected, the result is a false positive. False positives make security analysis more difficult by generating extra data points that need to be assessed, potentially clouding real incidents from detection.

False positives can be minimized through tuning of the faulty detection signatures—a process that should be performed regularly to ensure that detection devices are operating as efficiently as possible. While false positives often result in large amounts of unnecessary or irrelevant data, not all irrelevant data are false positives. Many security analysts and even security vendors are tempted to overly tune devices to eliminate any alert that occurs in large numbers because of this common misconception. The issue with overly aggressive tuning is that while it will make incidents easier to manage in day-to-day operations, it can introduce *false negatives*—that is, when a real threat fails to create an alert, or when a correlation rule fails to trigger because a necessary condition was suppressed by over-tuning (see Chapter 11, "Exception, Anomaly, and Threat Detection"). Remembering that event correlation signatures are signature-matching rules that detect known threat patterns, the elimination of smaller seemingly irrelevant events can prevent detection of the larger pattern. Similarly, as security researchers discover new patterns, event data that seem irrelevant today may become relevant in the future (see Figure 12.1).

To ensure accurate threat detection and correlation, all legitimately produced events should be retained short-term for live analysis (i.e. kept on-line) and long-term for forensic and compliance purposes (i.e. kept off-line) regardless of how irrelevant they may seem at the time of collection. Only true false positives—the events generated due to a false signature match—should be eliminated via tuning or filtering.

When considering the relevance of security events in industrial networks, consider the source of the event and its relevance to the specific zone being monitored. For example, all zones should have at least one perimeter security device, such as a firewall or IPS, but there may also be multiple host-based security devices capable of generating events, such as anti-virus, application whitelisting, intrusion detection and prevention systems (HIDS/HIPS), firewalls, or other security devices (see Chapter 9, "Establishing Zones and Conduits"). One example is industrial security appliances

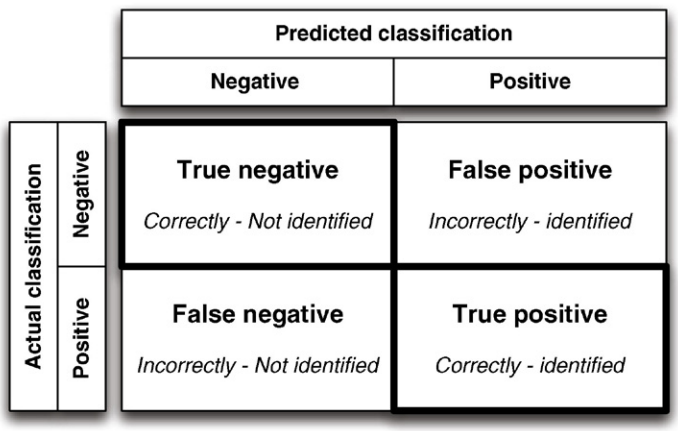

FIGURE 12.1 "Confusion Matrix" for event classification.

that use industrial protocol and application monitoring to enforce how industrial protocols are used.

These logs might provide much more specific data to a zone than do general security events, as seen in the example below from a Tofino industrial security appliance that provides detailed information pertaining to the unauthorized use of an industrial protocol (Modbus/TCP) function code (6 = "write single register"):

```
May 20 09:25:50 169.254.2.2 Apr 14 19:47:32 00:50:C2:B3:23:56
CEF:1|Tofino Security Inc|Tofino SA|02.0.00|300008|Tofino Modbus/
TCP Enforcer: Function Code List Check|6.0|msg = Function code 6
is not in permitted function code list TofinoMode = OPERATIONAL
smac = 9c:eb:02:a6:22 src = 192.168.1.126 spt = 32500
dmac = 00:00:bc:cf:6b:08 dst = 192.168.1.17 dpt = 502 proto = TCP
TofinoEthType = 800 TofinoTTL = 64 TofinoPhysIn = eth0
```

In contrast, a generic Snort IDS might produce a syslog event string identifying a perimeter policy violation, such as the attempted Windows update shown below, but cannot provide the context of application function codes within the industrial network (see Chapter 6, "Industrial Network Protocols").

```
Jan 01 00:00:00 [69.20.59.59] snort: [1:2002948:6] ET POLICY
External Windows Update in Progress [**] [Classification: Potential
Corporate Privacy Violation] [Priority: 1] {TCP} 10.1.10.33:1665
-> 192.168.25.35:80
```

An often-overlooked step prior to commissioning any device that will generate security events is to "tune" or validate that normal traffic does not trigger events. Figure 12.2 illustrates how a complete rule set for a Tofino Security Appliance might look once commissioned. Note that only the last rule (as indicated by the arrow) is actually enforcing segregation on the conduit by performing deep-packet inspection on Modbus/TCP (502/tcp) traffic originating in the ICS Host zone and destined for the ICS Controllers zone. There are many other types of valid traffic that is generated

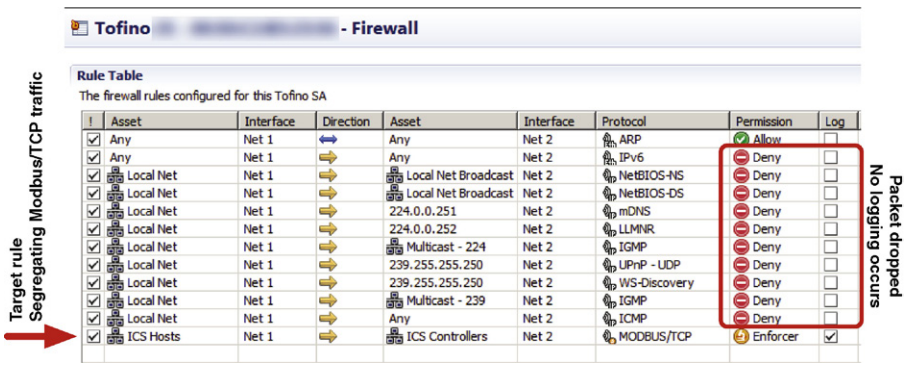

FIGURE 12.2 Tuning an industrial network security appliance.

to support functionality like the Network Neighborhood used in Windows operating systems and Neighboring Switches/Routers typical in both IT and OT network devices that is commonly sent to broadcast and multicast addresses. This valid traffic, if not properly handled with "drop-no log" entries in the rule set would generate "false positives" in terms of the security events within an industrial network. Some of the traffic that must be considered include

- Windows NetBIOS Traffic – Name Resolution Service (137/udp) and Datagram Server (138/udp)
- Multicast DNS (5353/udp)
- Link-Layer Multicast Name Resolution (5355/udp)
- Universal Plug 'n Play (1900/udp and 2869/tcp)
- Web Services Discovery Protocol (3702/udp)
- Cisco Discovery Protocol
- Link Layer Discovery Protocol
- Internet Control Message Protocol (IP Protocol 1)
- Internet Group Management Protocol (IP Protocol 2)
- Internet Protocol Version 6 (IPv6).

ASSETS

Assets—the physical devices connected to the network—also provide security data, typically in the form of logs. Assets can produce logs that track activity on a variety of levels. The operating system itself produces many logs, including system logs, application logs, and file system logs.

System logs are useful for tracking the status of devices and the services that are (or are not) running, as well as when patches are (or are not) applied. Logs are useful for determining the general health of an asset, as well as validating that approved ports and services are running. These logs are valuable in tracking which users (or applications) have authenticated to the asset, satisfying several compliance requirements. The following represents individual records from a Redhat Linux system log showing a successful user login, and a Windows failed authentication:

```
<345> Mar 17 11:23:15 localhost sshd[27577]: Accepted password
for knapp from ::ffff:10.1.1.1 port 2895 ssh2
<345> Fri Mar 17 11:23:15 2011 680 Security SYSTEM User Failure
Audit ENTERPRISE Account Logon attempt by:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon account: KNAPP Source
Workstation: ENTERPRISE Error Code: 0xC000006A 4574
```

Although syslog is ubiquitously used across a variety of systems, other event logging systems are used as well—the most notable of which is the Windows Management Instrumentation (WMI) framework. WMI produces auditable events in a structured data format that can be used against scripts (for automation) as well as by other Windows operating system functions.[5] Because syslog is so

widely supported, WMI events are often logged using a Windows syslog agent, such as Snare for Windows to stream WMI events over syslog. It is also possible to configure log forwarding between Windows hosts when restrictions prohibit the installation of agents on critical assets using the Windows Event Collector functionality.

The following WMI event example indicates the creation of a new process on a Windows server:

```
Computer Name: WIN-0Z6H21NLQO5
Event Code: 4688
Type: Audit Success (4)
User Name:
Category: Process Creation
Log File Name: Security
String[%1]: S-1-5-19
String[%2]: LOCAL SERVICE
String[%3]: NT AUTHORITY
String[%4]: 0x3e5
String[%5]: 0xc008
String[%6]: C:\Windows\System32\RacAgent.exe
String[%7]: %%1936
String[%8]: 0xc5e4
Message: A new process has been created. Subject: Security ID:
S-1-5-19 Account Name: LOCAL SERVICE Account Domain: NT AUTHORITY
Logon ID: 0x3e5 Process Information: New Process ID: 0xc008 New
Process Name: C:\Windows\System32\RacAgent.exe Token Elevation
Type: TokenElevationTypeDefault (1) Creator Process ID: 0xc5e4
Token Elevation Type indicates the type of token that was assigned
to the new process in accordance with User Account Control policy.
Type 1 is a full token with no privileges removed or groups
disabled. Afull token is only used if User Account Control is
disabled or if the user is the built-in Administrator account or
a service account. Type 2 is an elevated token with no privileges
removed or groups disabled. An elevated token is used when User
Account Control is enabled and the user chooses to start the
program using Run as administrator. An elevated token is also used
when an application is configured to always require administrative
privilege or to always require maximum privilege, and the user is
a member of the Administrators group. Type 3 is a limited token
with administrative privileges removed and administrative groups
disabled. The limited token is used when User Account Control is
enabled, the application does not require administrative privilege,
and the user does not choose to start the program using Run as
administrator.
```

The same event, when collected via syslog using a WMI agent, such as Snare, might look like this:

```
<12345> Fri Mar 17 11:23:15 2011||WIN-0Z6H21NLQ05||4688||Audit
Success (4)||||Process Creation||Security||S-1-5-19||LOCAL
SERVICE||NT AUTHORITY||0x3e5||0xc008||C:\Windows\System32\RacAgent.
exe||%%1936||0xc5e4
```

Application logs (covered in more detail under the section "Applications") provide a record of application-specific details, such as logon activities to an HMI, configuration changes, and other details that indicate how an application is being used. These Application Logs are an important component in the security associated with many ICS applications since these applications commonly utilize a single Windows logon authentication account and manage individual user actions via local application accounts and security settings.

File system logs typically track when files are created, changed, or deleted, when access privileges or group ownerships are changed, and similar details. File system logging is included in Windows using the Windows File Protection (WFP) within WMI, which is an "infrastructure for management data and operations on Windows-based operating systems."[6] File monitoring in Unix and Linux systems is performed using **auditd**, as well as with other commercial file integrity monitoring (FIM) products, such as Tripwire (www.tripwire.com) and nCircle (www.ncircle.com). These logs are extremely valuable for assuring the integrity of important files stored on an asset—such as configuration files (ensuring that the asset's configurations remain within policy), and the asset's log files themselves (ensuring that logged activities are valid and have not been tampered with to cover up indications of illicit behavior).

CONFIGURATIONS

Configuration monitoring refers to the process of monitoring baseline configurations for any indications of change,[7] and is only a small part of Configuration Management (CM). Basic configuration monitoring can be done at a rudimentary level through a combination of host configuration file monitoring (to establish the baseline), system and application log monitoring (to look for change actions), and FIM (to ensure that configurations are not altered). While this does not provide true CM, it does provide an indication as to when established configurations are altered, providing a valuable security resource.

Full CM systems provide additional key functions, typically mapping at least partially to the security controls outlined in NIST SP 800-53 under the section "Configuration Management," which provides a total of nine configuration management controls:[8]

- Configuration management policy and procedures—establishes a formal, documented configuration management policy.
- Baseline configurations—identifying and documenting all aspects of an asset's configurations to create a secure template against which all subsequent configurations are measured.

- Change control—monitoring for changes and comparing changes against the established baseline.
- Security impact analysis—the assessment of changes to determine and test how they might impact the security of the asset.
- Access restrictions for change—limiting configuration changes to a strict subset of administrative users.
- Configuration settings—identification, monitoring, and control of security configuration settings and changes thereto.
- Least functionality—the limitation of any baseline configuration to provide the least possible functionality to eliminate unnecessary ports and services.
- Information service (IS) component (asset) inventory—establishing an asset inventory to identify all assets that are subject to CM controls, as well as to detect rogue or unknown devices that may not meet baseline configuration guidelines.
- Establishment of a configuration management plan—assigning roles and responsibilities around an established CM policy to ensure that CM requirements are upheld.

Configuration management tools may also offer automated controls to allow batch configurations of assets across large networks, which is useful for ensuring that proper baselines are used in addition to improving desktop management efficiencies. For the purposes of security monitoring, it is the monitoring and assessment of the configuration files themselves that is a concern. This is because an attacker will often attempt to either escalate user privileges in order to obtain higher levels of access, or alter the configurations of security devices in order to penetrate deeper into secured zones—both of which are detectable with appropriate CM controls in place.

The logs produced by the CM are therefore a useful component of overall threat detection by using change events in combination with other activities, such as an event correlation system. For example, a port scan, followed by an injection attempt on a database, followed by a configuration change on the database server is indicative of a directed penetration attempt. Change logs are also highly beneficial (and in some cases mandatory) for compliance and regulatory purposes, with configuration and change management being a common requirement of most industrial security regulations (see Chapter 13, “Standards and Regulations”).

TIP

The problem with Configuration Management within ICS is that a large portion of the critical configuration information is retained in embedded devices often running proprietary or closed operating systems using nonstandard communication protocols. These devices (PLCs, RTUs, IEDs, SIS, etc.) represent the true endpoint with a connection to the physical process under control, making their configuration details (control logic, hardware configuration, firmware, etc.) one of the most critical components pertaining to the operational integrity of the ICS. While several available IT products, such as Tripwire, Solarwinds, and What’sUpGold, can provide configuration and change management for servers, workstations, and network devices, specialized products, such as Cyber Integrity™ by PAS and the Industrial Defender Automation Systems Manager from Lockheed Martin, provide not only the necessary database components to identify and track configuration changes, but an extensive library of system and device connectors necessary to extract configuration data from ICS components.

APPLICATIONS

Applications run on top of the operating system and perform specific functions. While monitoring application logs can provide a record of the activities relevant to those functions, direct monitoring of applications using a dedicated application monitoring product or application content firewall will likely provide a greater granularity of all application activities. Application logs can indicate when an application is executed or terminated, who logs into the application (when application-level security is implemented), and specific actions performed by users once logged in. The information contained in application logs is a summary, as it is in all log records. A sample application log record generated by an Apache web server is provided here:

```
Jan 01 00:00:00 [69.20.32.12] 93.80.237.221 - - [24/
Feb/2011:01:56:33 -0000] "GET/spambot/spambotmostseendownload.
php HTTP/1.0" 500 71224 "http://yandex.ru/yandsearch?text=video.
krymtel.net" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
MRA 4.6 (build 01425))"
```

A corresponding application log entry from an ICS illustrating a local access level change is shown here:

```
Jan 01 00:00:00 ICSSERVER1 HMI1 LEVEL Security Level Admin
Jan 01 00:00:00 ICSSERVER1 HMI1 LEVEL Security Level Oper
```

For a more detailed accounting of application activity, an application monitoring system can be used. For example, while it is possible that malware might be downloaded over HTTP, and be indicated in a log file, such as the first example shown earlier, monitoring an application's contents across a session could indicate malware that is embedded in a file being downloaded from an otherwise normal-seeming website, as shown in Figure 12.3.

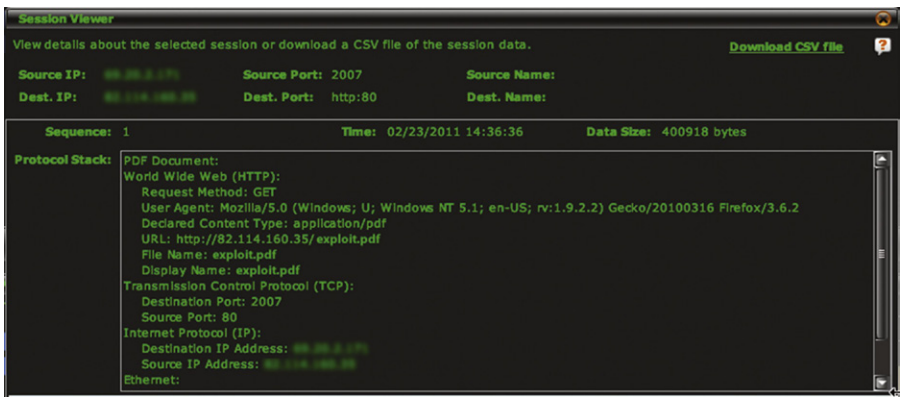

FIGURE 12.3 Application session details from an application monitor.

NETWORKS

Network flows are records of network communications, from a source to one or more destinations. Network infrastructure devices, such as switches and routers, usually track flows. Flow collection is typically proprietary to the network device manufacturer (e.g. Cisco supports NetFlow, and Juniper supports J-Flow), although many vendors also support the sFlow standard (see Table 12.1).

Table 12.1 Network Flow Details

Flow Detail	What It Indicates	Security Ramifications
SNMP interface indices (ifIndex in IF-MIB)	The size of the flow in terms of traffic volume (bytes, packets, etc.), as well as errors, latency, discards, physical addresses (MAC addresses), etc.	SNMP details can provide indications of abnormal protocol operation that might indicate a threat More germane to industrial networks, the presence of interface errors, latency, etc. can be directly harmful to the correct operation of many industrial protocols (see Chapter 6, "Industrial Network Protocols")
Flow start time	When a network communication was initiated and when it ended	Essential for the correlation of communications against security events
Flow end time	Collectively, the start and stop timestamps also indicate the duration of a network communications	
Number of bytes/ packets	Indicates the "size" of the network flow, indicative of how much data is being transmitted	Useful for the detection of abnormal network access, large file transfers, as might occur during information theft (e.g. retrieving a large database query result, downloading sensitive files, etc.)
Source and destination IP addresses	Indicates where a network communication began and where it was terminated	Essential for the correlation of related logs and security events (which often track IP address details)
Source and destination port	Note that in non-IP industrial networks, the flow may terminate at the IP address of an MI or PLC even though communications may continue over specialized industrial network protocols	IP addresses may also be used to determine the physical switch or router interface of the asset, or even the geographic location of the asset (through the use of a geo-location service)

Monitoring flows provides an overview of network usage over time (for trending analysis, capacity planning, etc.) as well as at any given time (for impact analysis, security assessment, etc.), and can be useful for a variety of functions, including[9]

- Network diagnosis and fault management.
- Network traffic management or congestion management.
- Application management, including performance management, and application usage assessments.
- Application and/or network usage accounting for billing purposes.
- Network security management, including the detection of unauthorized devices, traffic, and so on.

Network flow analysis is extremely useful for security analysis because it provides the information needed to trace the communications surrounding a security incident back to its source. For example, if an application whitelisting agent detects malware on an asset, it is extremely important to know where that malware came from, as it has already breached the perimeter defenses of the network and is now attempting to move laterally and infect adjacent machines. By correlating the malware attempt to network flows, it may be possible to trace the source of the malware and may also provide a path of propagation (i.e. where else did the virus propagate).

Network flow analysis also provides an indication of network performance for industrial network security. This is important because of the negative impact that network performance can have on process quality and efficiency, as shown in Table 12.1. An increase in latency can cause certain industrial protocols to fail, halting industrial processes.[10]

CAUTION

It is important to verify with the ICS supplier that network flow functionality can be enabled on the industrial network without negatively impacting the performance and integrity of the network and its connected devices. Many industrial protocols include real-time extensions (see Chapter 6, "Industrial Network Protocols") that see switch performance issues when available forwarding capacity has been altered. Network vendors like Cisco have addressed this with special "lite" capabilities for netflow reporting. Always consult the ICS supplier before making modifications to recommended or qualified network topologies and operating parameters.

USER IDENTITIES AND AUTHENTICATION

Monitoring users and their activities is an ideal method for obtaining a clear picture of what is happening on the network, and who is responsible. User monitoring is also an important component of compliance management, as most compliance regulations require specific controls around user privileges, access credentials, roles, and behaviors. This requirement is enforced more so on systems that must comply

with requirements, such as 21 CFR Part 11 and similar standards common in "FDA-regulated industries," such as pharmaceutical, food, and beverage.

Unfortunately, the term "user" is vague—there are user account names, computer account names, domain names, host names, and of course the human user's identity. While the latter is what is most often required for compliance management (see Chapter 13, "Standards and Regulations"), the former are what are typically provided within digital systems. Authentication to a system typically requires credentials in the form of a username and password, from a machine that has a host name, which might be one of several hosts in a named domain. The application itself might then authenticate to another backend system (such as a database), which has its own name and to which the application authenticates using yet another set of credentials. To further complicate things, the same human operator might need to authenticate to several systems, from several different machines, and may use a unique username on each. As mentioned earlier, ICS users may utilize a "common" Windows account shared by many, while each possesses a unique "application" account used for authentication and authorization within the ICS applications.

It is therefore necessary to normalize users to a common identity, just as it is necessary to normalize events to a common taxonomy. This can be done by monitoring activities from a variety of sources (network, host, and application logs), extracting whatever user identities might be present, and correlating them against whatever clues might be preset within those logs. For example, if a user authenticates to a Windows machine, launches an application and authenticates to it, and then the application authenticates to a backend system, it is possible to track that activity back to the original username by looking at the source of the authentications and the time at which they occurred. It can be assumed that all three authentications were by the same user because they occurred from the same physical console in clear succession.

As the systems become more complex and distributed, and as the number of users increases, each with specific roles and privileges, this can become cumbersome, and an automated identity management mechanism may be required.

This process is made simpler through the use of common directories, such as Microsoft Active Directory and/or the **Lightweight Directory Access Protocol** (LDAP), which act as identity directories and repositories. However, there may still be several unique sets of credentials per human operator that are managed locally within the applications versus centrally via a directory service. The difficulty lies in the lack of common log formats, and the corresponding lack of universal identities between diverse systems. User monitoring therefore requires the extraction of user information from a variety of network and application logs, followed by the normalization of that identity information. John Doe might log into a Windows domain using the username j.doe, have an e-mail address of jdoe@company.com, and log into a corporate intranet or Content Management System (CMS) as johnnyd, and so on. To truly monitor user behavior, it is necessary to recognize j.doe, jdoe, and johnnyd as a single identity.

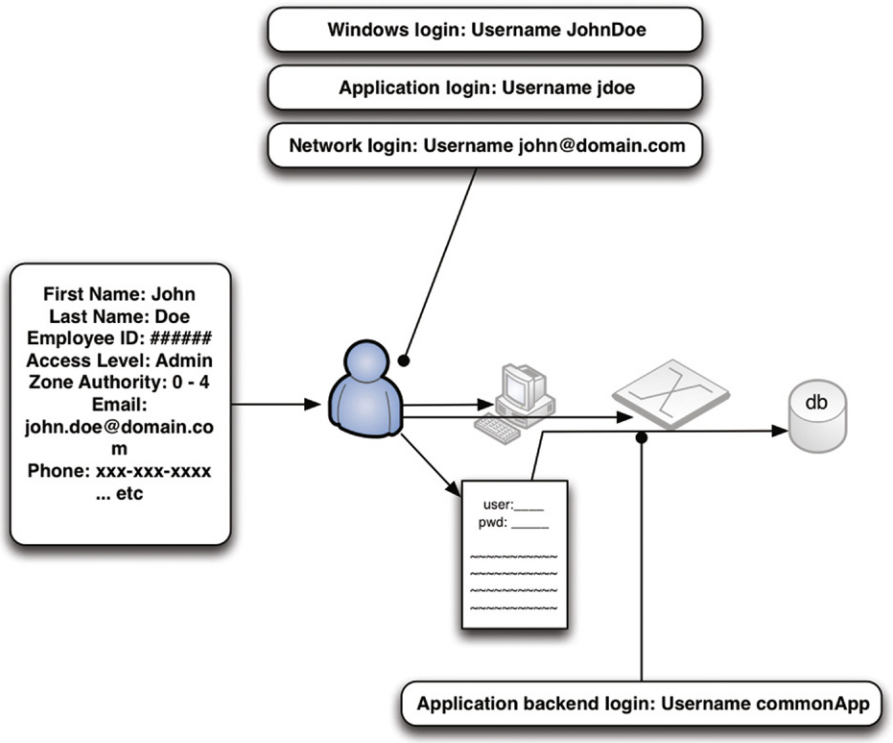

FIGURE 12.4 Normalization of user identity.

Several commercial identity and access management (IAM) systems (also sometimes referred to as identity and authentication management systems) are available to facilitate this process. Some commercially available IAM systems include: NetIQ (formerly Novell and spun off as part of the merger with Attachmate), Oracle Identity Management (also encompassing legacy Sun Identity Management prior to Oracle's acquisition of Sun Microsystems), and IBM's Tivoli Identity. Other third-party identity solutions, such as Securonix Identity Matcher, offer features of both a centralized directory and IAM by mining identity information from other IAMs and normalizing everything back to a common identity.[11] More sophisticated SIEM and Log Management systems might also incorporate identity correlation features to provide user normalization. An authoritative source of identity is provided by managing and controlling authentications to multiple systems via a centralized IAM irrespective of the method used, as shown in Figure 12.4.

Once the necessary identity context has been obtained, it can be utilized in the information and event management process to cross-reference logs and events back to users. A SIEM dashboard shows both network and event details associated with their source users in Figure 12.5.

FIGURE 12.5 User activity related to file access as displayed by an SIEM.

ADDITIONAL CONTEXT

While user identity is one example of contextual information, there is a wealth of additional information available that can provide context. This information—such as vulnerability references, IP reputation lists, and threat directories—supplements the monitored logs and events with additional valuable context. Examples of contextual information are provided in Table 12.2.

Contextual information is always beneficial, as the more context is available for any specific event or group of events, the easier it will be to assess relevance to specific security and business policies. This is especially true because the logs and events being monitored often lack the details that are most relevant, such as usernames (see Figure 12.6).[12]

It is important to know that contextual information adds to the total volume of information already being assessed. It is therefore most beneficial when used to enrich other security information in an automated manner (see section "Information Management").

BEHAVIOR

Behavior is not something that is directly monitored, rather it is the analysis of any monitored metric (obtained from a log, network flow, or other source) over time. The result is an indication of expected versus unexpected activity, which is extremely useful for a wide range of security functions, including anomaly-based threat detection, as well as capacity or threshold-based alarming. Behavior is also a useful condition in security event correlation (see Chapter 11, "Exception, Anomaly, and Threat Detection").

Behavior analysis is often provided by security log and event monitoring tools, such as log management systems, SIEMs, and network behavior anomaly detection

Table 12.2 Contextual Information Sources and Their Relevance

Information Source	Provided Context	Security Implications
Directory services (e.g. active directory)	User identity information, asset identity information, and access privileges	Provides a repository of known users, assets, and roles that can be leveraged for security threat analysis and detection, as well as for compliance
Identity and authentication management systems	Detailed user identity information, usernames and account aliases, access privileges, and an audit trail of authentication activity	Enables the correlation of users to access and activities based upon privilege and policy. When used to enrich security events, provides a clear audit trail of activity versus authority that is necessary for compliance auditing
Vulnerability scanner	Asset details including the operating system, applications in use (ports and services), patch levels, identified vulnerabilities, and related known exploits	Enables security events to be weighted based upon the vulnerability of their target (i.e. a Windows virus is less concerning if it is targeting a Linux workstation) Also provides valuable asset details for use in exception reporting, event correlation, and other functions
Penetration tester	Exploitation success/failure, method of exploitation, evasion techniques, etc.	Like with a vulnerability scanner, pen test tools provide the context of an attack vector. Unlike VA scan results, which show what could be exploited, a pen test indicates what has been exploited—which is especially useful for determining evasion techniques, detecting mutating code, etc.
Threat database/ CERT	Details, origins and recommendations for the remediation of exploits, malware, evasion techniques, etc. Threat intelligence may also be used as "watchlists," providing a cross-reference against which threats can be compared in order to highlight or otherwise call out threats of a specific category, severity, etc.	Threat intelligence can be used in a purely advisory capacity (e.g. providing educational data associated with a detected threat), or in an analytical capacity (e.g. in association with vulnerability scan data to weight the severity calculation of a detected threat)

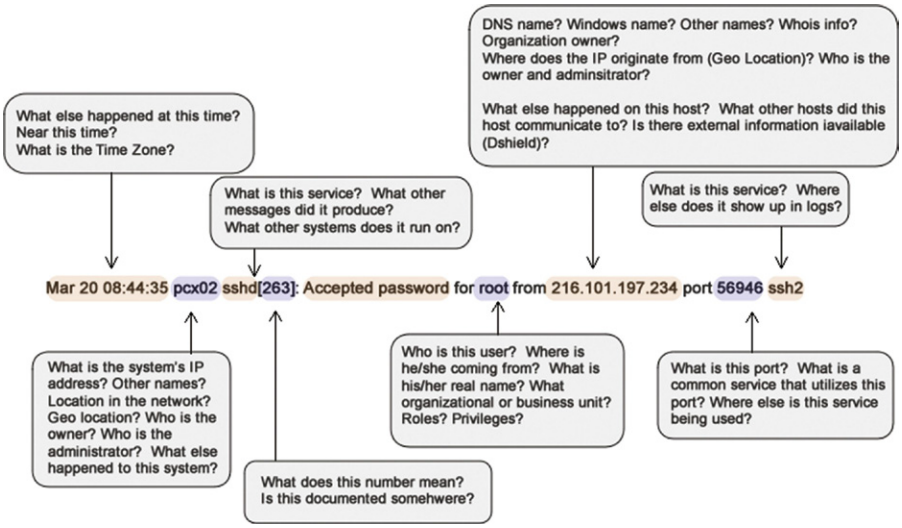

FIGURE 12.6 A log file, illustrating the lack of context image.

(NBAD) systems. If the system used for the collection and monitoring of security information does not provide behavioral analysis, an external tool, such as a spreadsheet or statistics program, may be required.

SUCCESSFULLY MONITORING SECURITY ZONES

Understanding what to monitor is only the first step—actually monitoring all of the users, networks, applications, assets, and other activities still needs to happen. The discussion of what to monitor focused heavily on logs, because log files are designed to describe activities that have occurred, are fairly ubiquitous, and are well understood. Log files are not always available however, and may not provide sufficient detail in some instances. Therefore, monitoring is typically performed using a combination of methods, including the following:

- Log collection and analysis
- Direct monitoring or network inspection
- Inferred monitoring via tangential systems.

Except in pure log-collection environments, where logs are produced by the assets and network devices that are already in place, specialized tools are required to monitor the various network systems. The results of monitoring (by whatever means) needs to be dealt with, because while manual logs and event reviews are possible (and allowed by most compliance regulations), automated tools are available and are recommended.

The central analysis of monitored systems is contrary to a security model built upon functional isolation. This is true because industrial networks should be separated into functional security zones, and centralized monitoring requires that log and event data either remain within a functional group (limiting the value for overall situation awareness of the complete system) or be shared between zones (potentially putting the security of the zone at risk). In the first scenario, logs and events are not allowed across the zone perimeter where they may be collected, retained, and analyzed only by local systems within that zone. In the second scenario, special considerations must be made for the transportation of log and event data across zone perimeters to prevent the introduction of a new inbound attack vector. A common method is to implement special security controls (such as a data diode, unidirectional gateway, or firewall configured to explicitly deny all inbound communications) to ensure that the security data are only allowed to flow toward the centralized management system. A hybrid approach may be used in industrial networks where critical systems in remote areas need to operate reliably. This provides local security event and log collection and management so that the zone can operate in total isolation, while also pushing security data to a central location to allow for more complete situational awareness across multiple zones.

LOG COLLECTION

Log collection is simply the collection of logs from whatever sources produce them. This is often a matter of directing the log output to a log aggregation point, such as a network storage facility and/or a dedicated Log Management system. Directing a log is often as simple as directing the syslog event data service to the IP address of the aggregator. In some cases, such as WMI, events are stored locally within a database rather than as log files. These events must be retrieved, either directly (by authenticating to Windows and querying the event database via the Windows Event Collector functionality) or indirectly (via a software agent, such as Snare, which retrieves the events locally and then transmits them via standard syslog transports).

DIRECT MONITORING

Direct monitoring refers to the use of a "probe" or other device to passively examine network traffic or hosts by placing the device in-line with the network. Direct monitoring is especially useful when the system being monitored does not produce logs natively (as is the case with many industrial network assets, such as RTUs, PLCs, and IEDs). It is also useful as a verification of activity reported by logs, as log files can be altered deliberately in order to hide evidence of malicious activities. Common monitoring devices include firewalls, intrusion detection systems (IDSs), **database activity monitors** (**DAMs**), application monitors, and network probes. These are often available commercially as software or appliances, or via open-source distributions, such as Snort (IDS/IPS), Wireshark (network sniffer and traffic analyzer), and Kismet (wireless sniffer).

Often, network monitoring devices produce logs of their own, which are then collected for analysis with other logs. Network monitoring devices are sometimes

referred to as "passive logging" devices because the logs are produced without any direct interaction with the system being monitored. Database activity monitors, for example, monitor database activity on the network—often on a span port or network tap. The DAM decodes network packets and then extracts relevant SQL transactions in order to produce logs. There is no need to enable logging on the database itself resulting in no performance impact to the database servers.

In industrial networks, it is similarly possible to monitor industrial protocol use on the network by providing "passive logging" to those industrial control assets that do not support logging. Passive monitoring is especially important in these networks, as many industrial protocols operate in real time and are highly susceptible to network latency and jitter. This is one reason why it is difficult to deploy logging agents on the devices themselves (which would also complicate asset testing policies), making passive network logging an ideal solution in these cases. Special consideration to any industrial network redundancy should also be considered when deploying network-based monitoring solutions.

In some instances, the device may use a proprietary log format or event streaming protocol that must be handled specially. Cisco's Security Device Event Exchange protocol (SDEE) (used by most Cisco IPS products) requires a username and password in order to authenticate with the security device so that events can be retrieved on demand, and/or "pushed" via a subscription model. While the end result is the same, it is important to understand that syslog is not absolutely ubiquitous.

INFERRED MONITORING

Inferred monitoring refers to situations where one system is monitored in order to infer information about another system. Many applications connect to a database. So as an example, monitoring the database in lieu of the application itself will provide valuable information about how the application is being used, even if the application itself is not producing logs or being directly monitored by an Application Monitor.

NOTE

Network-based monitoring inevitably leads to the question, "Is it possible to monitor encrypted network traffic?" Many industrial network regulations and guidelines recommend the encryption of control data when these data are transferred between trusted security zones via untrusted conduits … so how can these data be monitored via a network probe? There are a few options, each with benefits and weaknesses. The first is to monitor the sensitive network connection between the traffic source and the point of encryption. That is, encrypt network traffic externally using a network-based encryption appliance, such as the Certes Networks Enforcement Point (CEP) variable speed encryption appliances, and place the network probe immediately between the asset and the encryption. The second option is to utilize a dedicated network-based decryption device, such as the Netronome SSL Inspector. These devices perform deliberate, hardware-based man-in-the-middle attacks in order to break encryption and analyze the network contents for security purposes. A third option is not to monitor the encrypted traffic at all, but rather to monitor for instances of data that should be encrypted (such as industrial protocol function codes) but are not producing exception alerts indicating that sensitive traffic is not being encrypted.

To determine which tools are needed, start with your zone's perimeter and interior security controls (see Chapter 9, "Establishing Zones and Conduits") and determine which controls can produce adequate monitoring and which cannot. If they can, start by aggregating logs from the absolute perimeter (the demarcation between the least critical zone and any untrusted networks—typically the business enterprise LAN) to a central log aggregation tool (see the section "Information Collection and Management Tools"). Begin aggregating logs from those devices protecting the most critical zones, and work outward until all available monitoring has been enabled, or until the capacity of your log aggregation has become saturated. At this point, if there are remaining critical assets that are not being effectively monitored, it may be necessary to increase the capacity of the log aggregation system.

TIP

Adding capacity does not always mean buying larger, more expensive aggregation devices. Distribution is also an option—keep all log aggregation local within each zone (or within groups of similar zones), and then aggregate subsets of each zone to a central aggregation facility for centralized log analysis and reporting. While this type of event reduction will reduce the effectiveness of threat detection and will produce less comprehensive reports from the centralized system, all the necessary monitoring and log collection will remain intact within the zones themselves, where they can be accessed as needed.

This concept is particularly well-suited for industrial networks in that it allows the creation of a local "dashboard" where relevant events for nearby assets can be displayed and responded to quickly by a "first responder" that may reside in the operational or plant environment, while offering the ability to export these events to upper-level aggregators that have a much broader view of more assets, and can focus more on event correlation and threat analysis typically performed in a security operations center.

If all logs are being collected and there are still critical assets that are not adequately monitored, it may be necessary to add additional network monitoring tools to compensate for these deficiencies. This process is illustrated in Figure 12.7.

CAUTION

Remember that when aggregating logs it is still necessary to respect the boundaries of all established security zones. If logs need to be aggregated across zones (which is helpful for the detection of threats as they move between zones), make sure that the zone perimeter is configured to only allow the movement of logs in one direction; otherwise, the perimeter could potentially be compromised. In most instances, simply creating a policy that explicitly states the source (the device producing logs) and the destination (the log aggregation facility) for the specified service (e.g. syslog, port 514) is sufficient in order to enforce a restricted one-way transmission of the log files. For critical zones, physical separation using a data diode or unidirectional gateway may be required to assure that all log transmissions occur in one direction, and that there is no ability for malicious traffic to enter the secure zone from the logging facility.

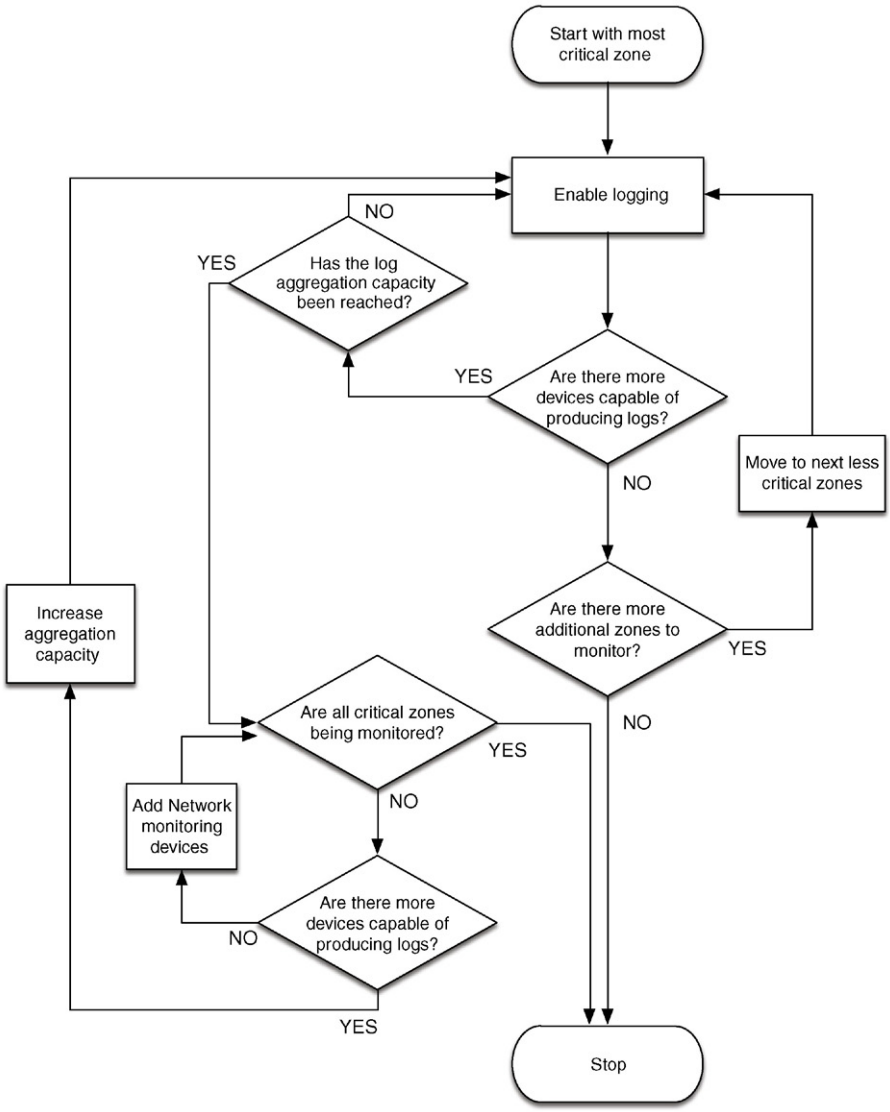

FIGURE 12.7 Process for enabling zone monitoring.

Additional monitoring tools might include any asset or network monitoring device, including host-based security agents, or external systems, such as an intrusion detection system, an application monitor, or an industrial protocol filter. Network-based monitoring tools are often easier to deploy, because they are by nature nonobtrusive and, if configured to monitor a spanned or mirrored interface, typically do not introduce latency.

INFORMATION COLLECTION AND MANAGEMENT TOOLS

The "log collection facility" is typically a log management system or a security information and event management (SIEM) system. These tools range from very simple to very complex and include free, open-source, and commercial options. Some options include syslog aggregation and log search, commercial log management systems, the open source security information management (**OSSIM**) system, and commercial security information and event management systems.

Syslog Aggregation and Log Search

Syslog allows log files to be communicated over a network. By directing all syslog outputs from supported assets to a common network file system, a very simple and free log aggregation system can be established. While inexpensive (essentially free), this option provides little added value in terms of utilizing the collected logs for analysis, requiring the use of additional tools, such as open source log search or IT search tools, or through the use of a commercial log management system or SIEM. If logs are being collected for compliance purposes as well as for security monitoring, additional measures will need to be taken to comply with log retention requirements. These requirements include nonrepudiation and chain of custody, as well as ensuring that files have not been altered, or accessed by unauthorized users. This can be obtained without the help of commercial systems, although it does require additional effort by IT managers.

Log Management Systems

Log management systems provide a commercial solution for log collection, analysis, and reporting. Log management systems provide a configuration interface to manage log collection, as well as options for the storage of logs—often allowing the administrator to configure log retention parameters by individual log source. At the time of collection, log management systems also provide the necessary nonrepudiation features to ensure the integrity of the log files, such as "signing" logs with a calculated hash that can be later compared to the files as a checksum. Once collected, the logs can then also be analyzed and searched, with the ability to produce prefiltered reports in order to present log data relevant to a specific purpose or function, such as compliance reports, which produce log details specific to one or more regulatory compliance controls, as shown in Figure 12.8.

Security Information and Event Management Systems

Security information and event management systems, or SIEMs, extend the capabilities of log management systems with the addition of specific analytical and contextual functions. According to security analysts from Gartner, the differentiating quality of an SIEM is that it combines the log management and compliance reporting qualities of a log management or legacy security information management (SIM) system with the real-time monitoring and incident management capabilities of a security event manager (SEM).[13] A SIEM must also support "data capture from heterogeneous data sources, including network devices, security devices, security programs,

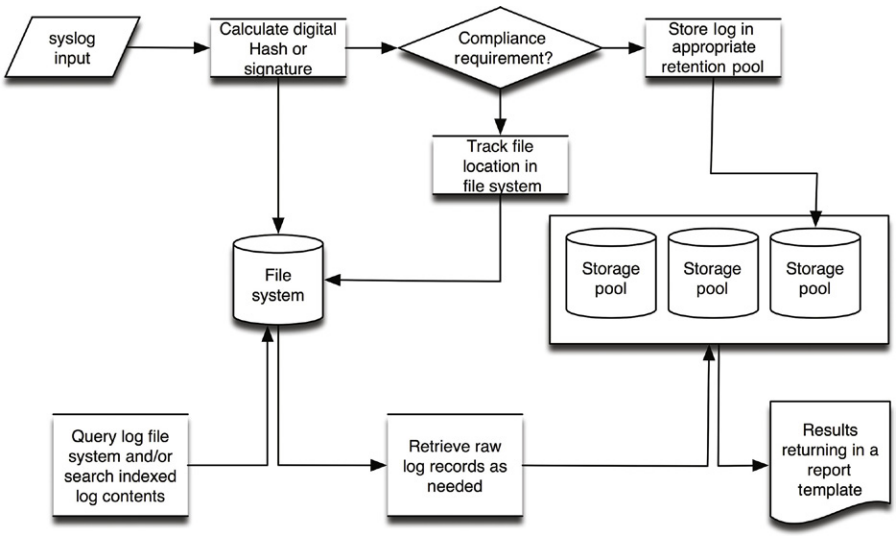

FIGURE 12.8 Typical log management operations.

and servers,"[14] making the qualifying SIEM an ideal platform for providing situational awareness across security zone perimeters and interiors.

Many SIEM products are available, including the open-source variants (OSSIM by AlienVault), as well as several commercial SIEMs (ArcSight by Hewlett-Packard, QRadar by IBM, LogRhythm, Enterprise Security Manager by McAfee, and Splunk Enterprise), competing across a variety of markets, and offering a variety of value-added features and specializations.

Because an SIEM is designed to support real-time monitoring and analytical functions, it will parse the contents of a log file at the time of collection, storing the parsed information in some sort of structured data store, typically a database or a specialized flat-file storage system. By parsing out common values, they are more readily available for analytics, helping to support the real-time goals of the SIEM, as shown in Figure 12.9. The parsed data are used for analytics, while a more traditional log management framework that will hash the logs and retain them for compliance. Because the raw log file may be needed for forensic analysis, a logical connection between the log file and the parsed event data is typically maintained within the data store.

SIEM platforms are often used in security operations centers (SOCs), providing intelligence to security operators that can be used to detect and respond to security concerns. Typically, the SIEM will provide visual dashboards to simplify the large amounts of disparate data into a more human-readable form. Figure 12.10 illustrates how a custom dashboard is created within Splunk to visual ICS-related security events. Figure 12.11 shows how this dashboard can be expanded to provide more application-layer event information pertaining to industrial protocol security events (e.g. use of invalid function codes).

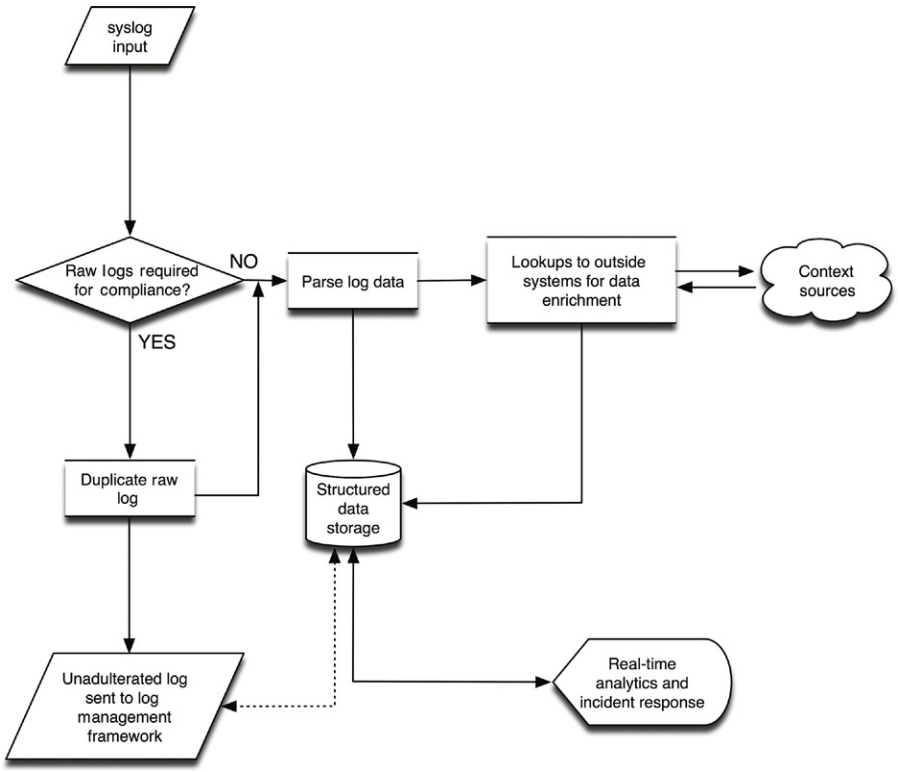

FIGURE 12.9 Typical SIEM operations.

NOTE

Log management and SIEM platforms are converging as information security needs become more closely tied to regulatory compliance mandates. Many traditional log management vendors now offer SIEM features, while traditional SIEM vendors are offering log management features.

Data Historians

Data Historians are not security monitoring products, but they do monitor activity (see Chapter 4, "Introduction to Industrial Control Systems and Operations") and can be a useful supplement to security monitoring solutions in several ways, including

- Providing visibility into control system assets that may not be visible to typical network monitoring tools.
- Providing process efficiency and reliability data that can be useful for security analysis.

Because most security monitoring tools are designed for enterprise network use, they are typically restricted to TCP- and UDP-based IP networks and therefore have

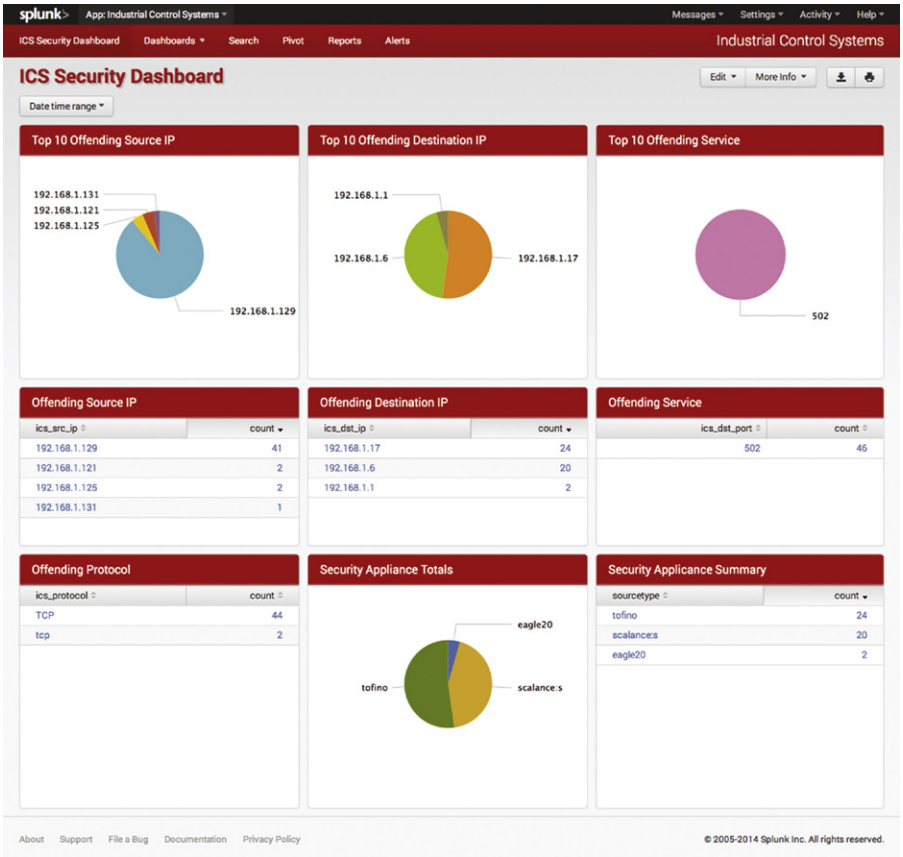

FIGURE 12.10 ICS security dashboard for Splunk.

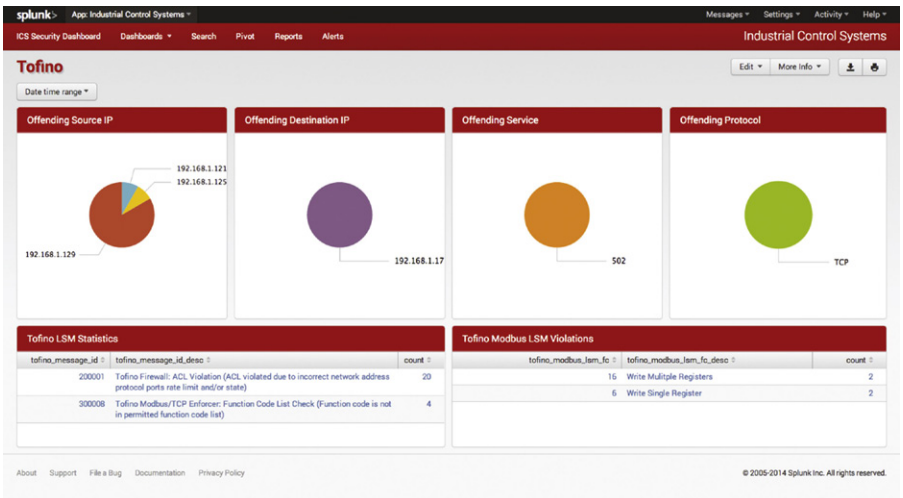

FIGURE 12.11 ICS security dashboard – application layer event analysis.

no visibility into large portions of most industrial plants that may utilize serial connectivity or other nonroutable protocols. Many industrial protocols are evolving to operate over Ethernet using TCP and UDP transports over IP, meaning these processes can be impacted by enterprise network activities. The security analysis capabilities of SIEM are made available to operational data by using the operational data provided by a Historian, allowing threats that originate in IT environments but target OT systems (i.e. Stuxnet and Dragonfly) to be more easily detected and tracked by security analysts. Those activities that could impact the performance and reliability of industrial automations systems can be detected as well by exposing IT network metrics to operational processes, including network flow activity, heightened latency, or other metrics that could impact the proper operation of industrial network protocols (see Chapter 6, "Industrial Network Protocols").

MONITORING ACROSS SECURE BOUNDARIES

As mentioned in the section "Successfully Monitoring Security Zones," it is sometimes necessary to monitor systems across secure zone boundaries via defined conduits. This requires zone perimeter security policies that will allow the security logs and events generated by the monitoring device(s) to be transferred to a central management console. Data diodes are ideal for this application as they force the information flow in one direction—away from the zones possessing higher security levels and toward the central management system. If a firewall is used, any "hole" provided for logs and events represents a potential attack vector. The configuration must therefore explicitly limit the communication from the originating source(s) to the destination management system, by IP (Layer 3), Port (Layer 4), and preferably application content (Layer 7), with no allowed return communication path. Ideally, this communication would be encrypted as well, as the information transmitted could potentially be sensitive in nature.

INFORMATION MANAGEMENT

The next step in security monitoring is to utilize the relevant security information that has been collected. Proper analysis of this information can provide the situational awareness necessary to detect incidents that could impact the safety and reliability of the industrial network.

Ideally, the SIEM or Log Manager will perform many underlying detection functions automatically—including normalization, data enrichment, and correlation (see Chapter 11, "Exception, Anomaly, and Threat Detection")—providing the security analyst with the following types of information at their disposal:

- The raw log and event details obtained by monitoring relevant systems and services, normalized to a common taxonomy.

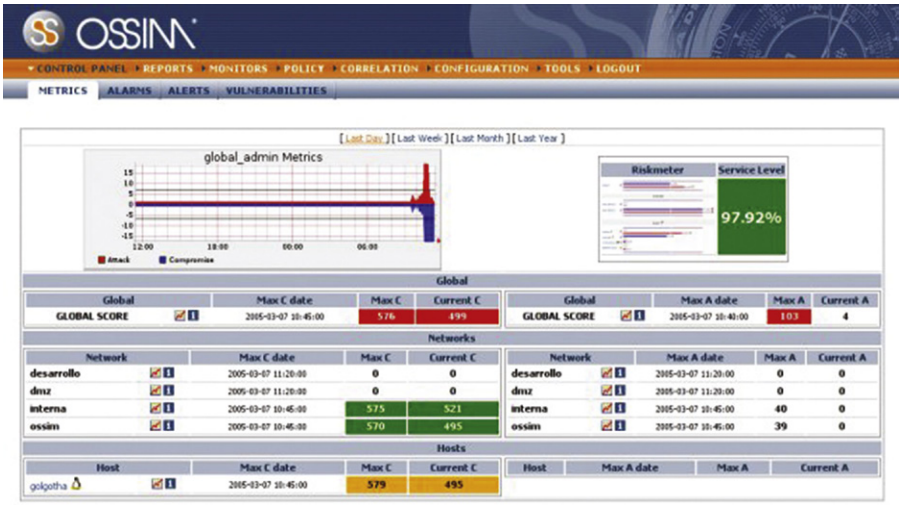

FIGURE 12.12 The Open Source Security Information Management project.

- The larger "incidents" or more sophisticated threats derived from those raw events that may include correlation with external global threat intelligence sources.
- The associated necessary context to what has been observed (raw events) and derived (**correlated events**).

Typically, an SIEM will represent a high-level view of the available information on a dashboard or console, as illustrated in Figure 12.12, which shows the dashboard of the Open Source Security Information Management (OSSIM) platform. With this information in hand, automated and manual interaction with the information can occur. This information can be queried directly to achieve direct answers to explicit questions. It can also be formulated into a report to satisfy specific business, policy, or compliance goals, or it can be used to proactively or reactively notify a security or operations officer of an incident. The information is available to further investigate incidents that have already occurred.

QUERIES

The term "query" refers to a request for information from the centralized data store. This can sometimes be an actual database query, using structured query language (SQL), or it may be a plain-text request to make the information more accessible by users without database administration skills (although these requests may use SQL queries internally, hidden from the user). Common examples of initial queries include the following:

- Top 10 talkers (by total network bandwidth used)
- Top talkers (by unique connections or flows)

- Top events (by frequency)
- Top events (by severity)
- Top events over time
- Top applications in use
- Open ports.

These requests can be made against any or all data that are available in the data store (see the section "Data Availability"). By providing additional conditions or filters, queries can be focused yielding results more relevant to a specific situation. For example

- Top 10 talkers during non-business hours
- Top talkers using specific industrial network protocols
- All events of a common type (e.g. user account changes)
- All events targeting a specific asset or assets (e.g. critical assets within a specific zone)
- All ports and services used by a specific asset or assets
- Top applications in use within more than one zone.

Query results can be returned in a number of ways: via delimited text files, a graphical user interface or dashboard, preformatted executive reports, an alert that is delivered by SMS or e-mail, and so on. Figure 12.13 shows user activity filtered by a specific event type—in this example, administrative account change activities that correspond with NERC compliance requirements.

A defining function of an SIEM is to correlate events to find larger incidents (see Chapter 11, "Exception, Anomaly, and Threat Detection"). This includes the ability to define correlation rules, as well as present the results via a dashboard. Figure 12.14 shows a graphical event correlation editor that allows the logical conditions (such as

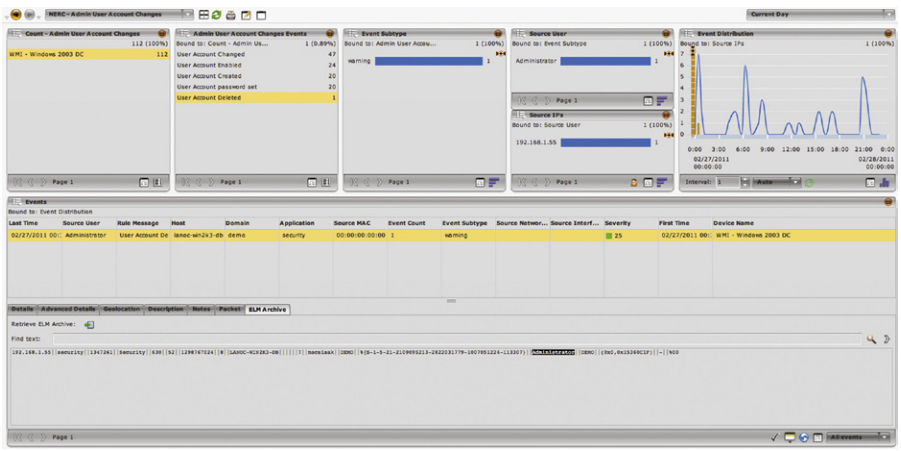

FIGURE 12.13 An SIEM dashboard showing administrative account changes.

FIGURE 12.14 An example of a graphical interface for creating event correlation rules.

"if A and B then C"), while Figure 12.15 shows the result of an incident query—in this case the selected incident (an HTTP Command and Control Spambot) being derived from four discrete events.

REPORTS

Reports select, organize, and format all relevant data from the enriched logs and events into a single document. Reports provide a useful means to present almost any data set. Reports can summarize high-level incidents for executives, or include precise and comprehensive documentation that provides minute details for internal auditing or for compliance. An example of a report generated by an SIEM is shown in Figure 12.16 showing a quick summary of the OSIsoft PI Historian authentication failures and point change activity.

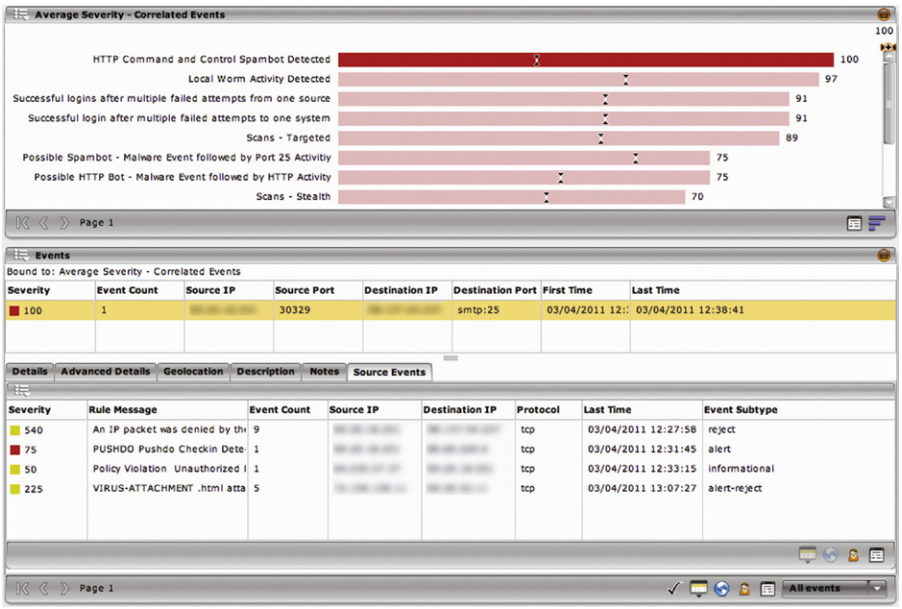

FIGURE 12.15 An SIEM dashboard a correlated event and its source events.

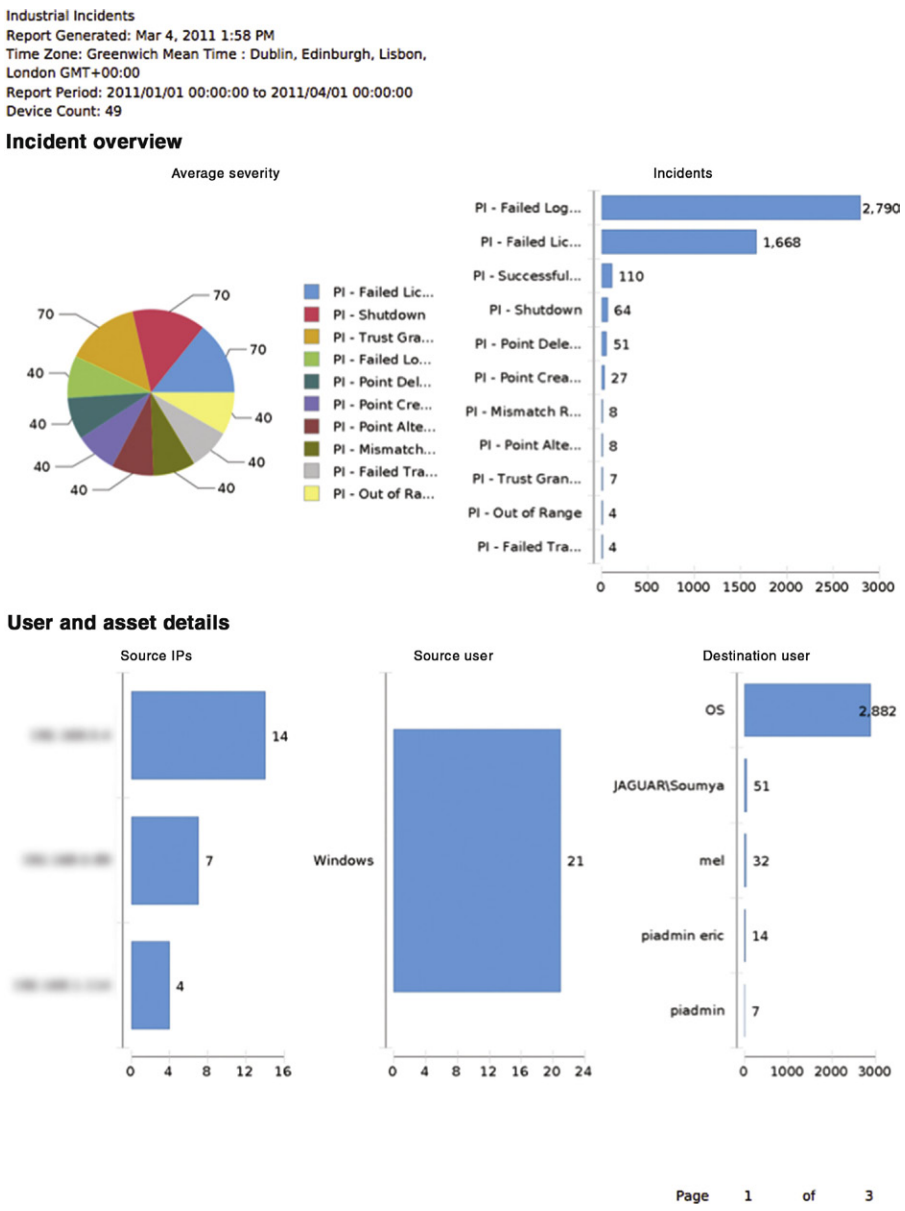

FIGURE 12.16 An SIEM report showing industrial activities.

ALERTS

Alerts are active responses to observed conditions within the SIEM. An alert can be a visual notification in a console or dashboard, a direct communications (e-mail, page, SMS, etc.) to a security administrator, or even the execution of a custom script. Common alert mechanisms used by commercial SIEMs include the following:

- Visual indicators (e.g. red, orange, yellow, green)
- Direct notification to a user or group of users
- Generation and delivery of a specific report(s) to a user or group of users
- Internal logging of alert activity for audit control
- Execution of a custom script or other external control
- Generation of a ticket in a compatible help desk or incident management system.

Several compliance regulations, including NERC CIP, CFATS, and NRC RG 5.71, require that incidents be appropriately communicated to proper authorities inside and/or outside of the organization. The alerting mechanism of an SIEM can facilitate this process by creating a useable variable or data dictionary with appropriate contacts within the SIEM and automatically generating appropriate reports and delivering them to key personnel.

INCIDENT INVESTIGATION AND RESPONSE

SIEM and log management systems are useful for incident response, because the structure and normalization of the data allow an incident response team to drill into a specific event to find additional details (often down to the source log file contents and/or captured network packets), and to pivot on specific data fields to find other related activities. For example, if there is an incident that requires investigation and response, it can be examined quickly providing relevant details, such as the username and IP address. The SIEM can then be queried to determine what other events are associated with the user, IP, and so on.

In some cases the SIEM may support active response capabilities, including

- Allowing direct control over switch or router interfaces via SNMP, to disable network interfaces.
- Executing scripts to interact with devices within the network infrastructure, to reroute traffic, isolate users, and so on.
- Execute scripts to interact with perimeter security devices (e.g. firewalls) to block subsequent traffic that has been discovered to be malicious.
- Execute scripts to interact with directory or IAM systems to alter or disable a user account in response to observed malicious behavior.

These responses may be supported manually or automatically, or both.

CAUTION

While automated response capabilities can improve efficiencies, they should be limited to non-critical security zones and/or to zone perimeters. As with any control deployed within industrial networks, all automated responses should be carefully considered and tested prior to implementation. A false positive could trigger such a response and cause the failure of an industrial operation, with potentially serious consequences.

LOG STORAGE AND RETENTION

The end result of security monitoring, log collection, and enrichment is a large quantity of data in the form of log files, which must be stored for audit and compliance purposes (in the cases where direct monitoring is used in lieu of log collection, the monitoring device will still produce logs, which must also be retained). This represents a few challenges, including how to ensure the integrity of the stored files (a common requirement for compliance), how and where to store these files, and how they can be kept readily available for analysis.

NONREPUDIATION

Nonrepudiation refers to the process of ensuring that a log file has not been tampered with, so that the original raw log file can be presented as evidence, without question of authenticity, within a court of law. This can be achieved in several ways, including digitally signing log files upon collection as a checksum, utilizing protected storage media, or the use of third-party FIM systems.

A digital signature is typically provided in the form of a hash algorithm that is calculated against the log file at the time of collection. The result of this calculation provides a checksum against which the files can be verified to ensure they have not been tampered with. If the file is altered in any way, the hash will calculate a different value and the log file will fail the integrity check. If the checksum matches, the log is known to be in its original form.

The use of appropriate storage facilities can ensure nonrepudiation as well. For example, by using write once read many (WORM) drives, raw log records can be accessed but not altered, as the write capability of the drive prevents additional saves. Many managed storage area network (SAN) systems also provide varying levels of authentication, encryption, and other safeguards.

A FIM may already be in use as part of the overall security monitoring infrastructure, as described in the section “Assets.” The FIM observes the log storage facility for any sign of changes or alterations, providing an added level of integrity validation.

DATA RETENTION/STORAGE

The security monitoring tools just mentioned all require the collection and storage of security-related information. The amount of information that is typically required

could easily surpass 170 GB over an 8-h period for a medium-sized enterprise collecting information at approximately 20,000 events per second.[15] It is worth mentioning that event generation within an industrial network is typically a small fraction of this number, and when properly tuned, presents a manageable amount of information storage.

Data retention refers to the amount of information that is stored long-term, and can be measured in volume (the size of the total collected logs in bytes) and time (the number of months or years that logs are stored for). The length of time a log is retained is important, as this metric is often defined by compliance regulations—NERC CIP requires that logs are retained for anywhere from 90 days to up to 3 years, depending upon the nature of the log.[16] The amount of physical storage space that is required can be calculated by determining which logs are needed for compliance and for how long they must be kept. Some of the factors that should be considered include the following:

- Identifying the quantity of inbound logs
- Determining the average log file size
- Determining the period of retention required for logs
- Determining the supported file compression ratios of the log management or SIEM platform being used.

Table 12.3 illustrates how sustained log collection rates map to total log storage requirements over a retention period of 7 years, resulting in a few terabytes (10^{12}) of storage up to hundreds of terabytes or even petabytes (10^{15}) of storage.

There may be a requirement to retain an audit trail for more than one standard or regulation depending upon the nature of the organization, often with each regulation mandating different retention requirements. As with NERC CIP, there may also be a change in the retention requirements depending upon the nature of the log, and whether an incident has occurred. All of this adds up to even greater, long-term storage requirements.

Table 12.3 Log Storage Requirements Over Time

Logs per Second	Logs per Day (in Billions)	Logs per Year (in Billions)	Average Bytes per Event	Retention Period in Years	Raw Log Size (TB)	Compressed Bytes (TB) 5:1	Compressed Bytes (TB) 10:1
100,000	8.64	3154	508	7	10,199	2040	1020
50,000	4.32	1577	508	7	5,100	1020	510
25,000	2.16	788	508	7	2,550	510	255
10,000	0.86	315	508	7	1,020	204	102
5,000	0.43	158	508	7	510	102	51
1,000	0.09	32	508	7	102	21	11
500	0.04	16	508	7	51	11	6

TIP

Make sure that the amount of available storage has sufficient headroom to accommodate spikes in event activity, because event rates can vary (especially during a security incident).

DATA AVAILABILITY

Data availability differs from retention, referring to the amount of data that is accessible for analysis. Also called "live" or "online" data, the total data availability determines how much information can be analyzed concurrently—again, in either volume (bytes and/or total number of events) or time. Data retention affects the ability of an SIEM to detect "low and slow" attacks (attacks that purposefully occur over a long period of time in order to evade detection), as well as to perform trend analysis and anomaly detection (which by definition requires a series of data over time—see Chapter 11, "Exception, Anomaly, and Threat Detection").

TIP

In order to meet compliance standards, it may be necessary to produce a list of all network flows within a particular security zone that originated from outside of that zone, for the past 3 years. For this query to be successful, 3 years of network flow data need to be available to the SIEM at once. There is a work-around if the SIEM's data availability is insufficient (for example, it can only keep 1 year of data active). The information can be stored in volumes consistent with the SIEM's data availability by archiving older data sets. A partial result is obtained by querying the active data set. Two additional queries can be run by then restoring the next-previous backup or archive, producing multiple partial result sets of 1 year each. These results can then be combined to obtain the required 3-year report. Note that this requires extra effort on the part of the analyst. The archive/retrieval process on some legacy SIEMs may interfere with or interrupt the collection of new logs until the process is complete.

Unlike data retention, which is bound by the available volume of data storage (disk drive space), data availability is dependent upon the structured data that are used by the SIEM for analysis. Depending upon the nature of the data store, the total data availability of the system may be limited to a number of days, months, or years. Typically, one or more of the following limits databases:

- The total number of columns (indices or fields)
- The total number of rows (discreet records or events)
- The rate at which new information is inserted (i.e. collection rate)
- The rate at which query results are required (i.e. retrieval rates).

Depending upon the business and security drivers behind information security monitoring, it may be necessary to segment or distribute monitoring and analysis into zones to meet performance requirements. Some factors to consider when calculating the necessary data availability include

- The total length of time over which data analysis may be required by compliance standards.

- The estimated quantity of logs that may be collected in that time based on event estimates.
- The incident response requirements of the organization—certain governmental or other critical installations may require rapid-response initiatives that necessitate fast data retrieval.
- The desired granularity of the information that is kept available for analysis (i.e. are there many vs. few indices).

SUMMARY

A larger picture of security-related activity begins to form once zone security measures are in place. Exceptions from the established security policies can then be detected by measuring these activities and further analyzing them. Anomalous activities can also be identified so that they may be further investigated.

This requires well-defined policies with those policies configured within an appropriate information analysis tool. Just as with perimeter defenses to the security zone, carefully built variables defining allowed assets, users, applications, and behaviors can be used to aid in detection of security risks and threats. If these lists can be determined dynamically, in response to observed activity within the network, the "whitelisting" of known-good policies, becomes "smart-listing." This helps further strengthen perimeter defenses through dynamic firewall configuration or IPS rule creation.

The event information can be further analyzed as various threat detection techniques are used together by event correlation systems that find larger patterns more indicative of serious threats or incidents. Widely used in IT network security, event correlation is beginning to "cross the divide" into OT networks, at the heels of Stuxnet and other sophisticated threats that attempt to compromise industrial network systems via attached IT networks and services.

Everything (measured metrics, baseline analysis, and whitelists) rely on a rich base of relevant security information. Where does this security information come from? The networks, assets, hosts, applications, protocols, users, and everything else that is logged or monitored contributes to the necessary base of data required to achieve "situational awareness" and effectively secure an industrial network.

ENDNOTES

1. J.M. Butler. Benchmarking Security Information Event Management (SIEM). The SANS Institute Analytics Program, February, 2009.
2. Ibid.
3. Ibid.
4. Ibid.
5. Microsoft. Windows Management Instrumentation. <http://msdn.microsoft.com/en-us/library/aa394582(v=VS.85).aspx>, January 6, 2011 (cited: March 3, 2011).

6. Ibid.
7. National Institute of Standards and Technology, Special Publication 800-53 Revision 3. Recommended Security Controls for Federal Information Systems and Organizations, August, 2009.
8. Ibid.
9. Flow.org. Traffic Monitoring using sFlow. <http://www.sflow.org/sFlowOverview.pdf>, 2003 (cited: March 3, 2011).
10. B. Singer, Kenexis Security Corporation, in: D. Peterson (Ed.), Proceedings of the SCADA Security Scientific Symposium, 2: Correlating Risk Events and Process Trends to Improve Reliability, Digital Bond Press, 2010.
11. Securonix, Inc., Securonix Indentity Matcher: Overview. <http://www.securonix.com/identity.htm>, 2003 (cited: March 3, 2011).
12. A. Chuvakin, Content Aware SIEM. <http://www.sans.org/security-resources/idfaq/vlan.php> February, 2000 (cited: January 19, 2011).
13. K.M. Kavanagh, M. Nicolett, O. Rochford, "Magic quadrant for security information and event management," Gartner Document ID Number: G00261641, June 25, 2014.
14. Ibid.
15. J.M. Butler, Benchmarking Security Information Event Management (SIEM). The SANS Institute Analytics Program, February, 2009.
16. North American Electric Reliability Corporation. NERC CIP Reliability Standards, version 4. <http://www.nerc.com/page.php?cid=2|20> February 3, 2011 (cited: March 3, 2011).