CHAPTER

Exception, Anomaly, and Threat Detection

11

INFORMATION IN THIS CHAPTER

- Exception Reporting
- Behavioral Anomaly Detection
- Behavioral Whitelisting
- Threat Detection

Clear policies about what communications are allowed and what are not have already been obtained by defining zones. The operation within each zone should also be well defined and relatively predictable. This supports two important types of behavioral analysis: exception reporting and anomaly detection.

Exception reporting refers to an automated system that notifies the security administrator whenever a defined policy has been violated. In the context of zone-based security, this means a notification that the defined zone has been violated—a user, system, or service is interacting with the zone in a way that is contrary to security policies established at the perimeter and/or within the zone interior (see Chapter 9, "Establishing Zones and Conduits"). If we expect one behavior but see another, we can view this behavior as a potential threat and take action accordingly.

Anomaly detection picks up where policy-based detection ends, by providing a "rule-less" method of identifying possible threat behavior. Anomaly detection simply takes action when something out of the ordinary occurs. In an industrial system—especially if a strong defense-in-depth posture is maintained and zones are appropriately separated—the normal behavior can be determined, and variations in that behavior should be minimal. The operational behavior of an industrial network should be relatively predictable making anomaly detection effective once all "normal" actions have been defined.

The effectiveness of anomaly detection pivots on that basic understanding of behavior. Understanding how baseline behavior can be measured is the first step to implementing a usable anomaly detection strategy.

Taken together, clearly defined policies and anomaly detection can provide an additional function called Behavioral Whitelisting. Behavioral Whitelisting combines an understanding of what is known good/bad behavior (policies) with an understanding of expected behaviors, to define what is "known good behavior." Just as whitelists of other known good elements (IP addresses, applications, users, etc.) can be used to enforce perimeter and interior zone defenses, these higher level behavioral whitelists can help to deter broader threats, even across zones.

Although each method is effective on its own, attacks rarely occur in clear, direct paths (see Chapter 8 "Risk and Vulnerability Assessments"). Therefore, to detect more sophisticated threats, all anomalies and exceptions need to be assessed together, along with the specific logs and events generated by network switches, routers, security appliances, and other devices including critical industrial control system (ICS) Windows-based assets. Event correlation looks across all systems to determine larger threat patterns that can more clearly identify a security incident. Event correlation is only as good as the data that are available, requiring that all of the mentioned detection techniques be used to generate a comprehensive base of relevant security information. It also requires proper monitoring of networks and devices, as discussed in the next chapter, "Security Monitoring of Industrial Control Systems".

CAUTION

Automated tools for the detection of exceptions, anomalies, and advanced threats are effective measures to help notify security analysts of incidents that may need to be addressed. However, no tool should be trusted completely; the experience and insight of a human analyst is a valuable component in the security monitoring and analysis process. While tools are often sold with the promise of being "an analyst in a box," even the most well-tuned systems will still produce false positives and false negatives, therefore requiring the additional layer of human intellect to complete the assessment. At the time of publishing, several credible companies have begun offering ICS-focused Managed Security Services that can provide the much needed 24×7 security coverage to industrial networks that is absent from many production environments today.

EXCEPTION REPORTING

In Chapter 9, "Establishing Zones and Conduits," specific policies have been developed and enforced by firewalls, intrusion prevention systems, application monitors, and other security devices. Apart from the clear examples of when a specific firewall or intrusion prevention system (IPS) rule triggers an alert, these policies can be used to assess a variety of behaviors. Exception reporting looks at all behaviors, and unlike a hard policy defined on the conduits at a zone's perimeter, which makes black-and-white decisions about what is good and bad, exception reporting can detect suspicious activities by compiling a wealth of seemingly benign security events.

This level of assessment could encompass any measurable function of a zone(s), including network traffic patterns, user access, and operational controls. At a very basic level, exception reporting might be used to inform an operator when something that should not have been allowed (based on zone perimeter policies) has occurred. The first example in Table 11.1 illustrates the concept that it should not be possible for inbound network communication to originate from an unrecognized IP address—that should have been prevented by the default Deny All firewall policy.

Other less obvious uses for exception reporting are exemplified in the last example in Table 11.1, where two completely different detection methods (an application monitoring system and a log analysis system) indicate a policy exception that

Table 11.1 Examples of Suspicious Exceptions

Exception	Policy being Enforced	Detected by	Recommended Action
Network flow originates from a different zone than the destination IP address	Network separation of functional groups/zones	Firewall, Network Monitor, Network IDS/IPS, etc. using $Zone_IP variables	Alert only, to create a report on all inter-zone communications
Network traffic originating from foreign IP addresses is seen within a secured zone	Isolation of critical zones from the Internet or Outside addresses	Log Manager/Analyzer, SIEM, etc. correlating !$Zone_IP variables and geolocation data	Critical Alert to indicate possible penetration of a secure zone
Authorized user accessing the network from a new or different IP address	User access control policies	Log Manager/Analyzer, SIEM, etc. correlating $Zone_IP variables to user authentication activity	Alert only, to create a report on abnormal administrator activity
Unauthorized user performing administrator functions	User access control policies	Log Manager/Analyzer, SIEM, etc. correlating !$Admin_users variables to application activity	Critical Alert to indicate potential unauthorized privilege escalation
Industrial protocol used in nonindustrial zones	Network separation of functional groups by protocol	Network Monitor, Network IDS/IPS, Application Monitor, Industrial Protocol Monitor, etc. using !$Zone_Protocol variables	Alert only, to create a report of abnormal protocol use
Industrial Protocol using WRITE function codes outside of normal business hours	Administrative control policies	Application monitoring detects $Modbus_Administrator_Functions	Alert only, to create an audit trail of unexpected admin behavior
Identity or authentication systems indicate normal administrative shifts			
SIEM or other log analysis tool correlates administrative functions against expected shift hours			

(Continued)

Table 11.1 Examples of Suspicious Exceptions *(cont.)*

Exception	Policy being Enforced	Detected by	Recommended Action
Industrial protocol using WRITE function codes is originating from a device authenticated to a nonadministrative user Authentication logs indicate a nonadministrative user SIEM or other log analysis tool correlates authentication logs with control policies and industrial protocol functions	User access control policies	Application monitoring detects $Modbus_Administrator_Functions	Critical Alert to indicate possible insider threat or sabotage

otherwise might seem benign. In this example, the function codes in question are only a concern if executed by an unauthorized user.

Exception reporting can be automated using many log analysis or security information management systems, which are designed to look at information (typically log files) from many sources, and correlate this information together (for more information on how to generate this information, see Chapter 12, "Security Monitoring of Industrial Control Systems"). Exceptions cannot be determined without an understanding of the policies that are in place. Over time, exception reporting should evolve, such that fewer exceptions occur—and therefore fewer reports—as the process matures.

BEHAVIORAL ANOMALY DETECTION

Sometimes, an exception might be seen in a network's expected behavior, rather than in adherence to policy. These anomalies can be detected by comparing monitored behavior against known "normal" values. This can be done in a variety of ways: manually, based on real-time monitoring; manually, via log review; automatically, using a network behavior anomaly detection (NBAD) product, log analysis, or security information and event management (SIEM) tool; or automatically, by exporting data to a dedicated spreadsheet or other statistical application. Note that even with highly automated systems—such as SIEM—a degree of human analysis is still required. The value of an automation tool is in its ability to simplify the process for the human analyst, using various detection algorithms, correlation, event scoring, and other techniques to add context to the raw data. Beware of any tool that claims to eliminate the need for human cognizance, as there is no such thing as an "analyst in a box." Whether performed manually or automatically, an anomaly cannot be detected without an

established baseline of activity upon which to compare. Once a baseline has been established for a given metric (such as the volume of network traffic and the number of active users), that metric must be monitored using one or more of the methods described in Chapter 12, "Security Monitoring of Industrial Control Systems."

MEASURING BASELINES

Baselines are time-lagged calculations based on running averages. They provide a basis (base) for comparison against an expected value (line). Baselines are useful for comparing past behaviors to current behaviors, but can also be used to measure network or application capacity, or almost any other operational metric that can be tracked over time. A baseline should not be confused with a trend analysis—a baseline is a value; nothing more, nothing less. Using that metric in an analysis of past-observed behavior and future-predicted behavior is a trend analysis—a forward-looking application of known baselines to predict the continuation of observed trends.

A baseline can be simple or complex—anything from a gut understanding of how a system works to a sophisticated statistical calculation of hard, quantifiable data. The simplest method of establishing a baseline is to take all data collected over a period of time and use whatever metric is available to determine the average over time. This is a commonly used method that is helpful in determining whether something is occurring above or below a fixed level. In Figure 11.1, for example, it can be clearly seen that production output is either above or below the average production level for the previous 12 months. The specific peaks and valleys could represent anything from a stalled

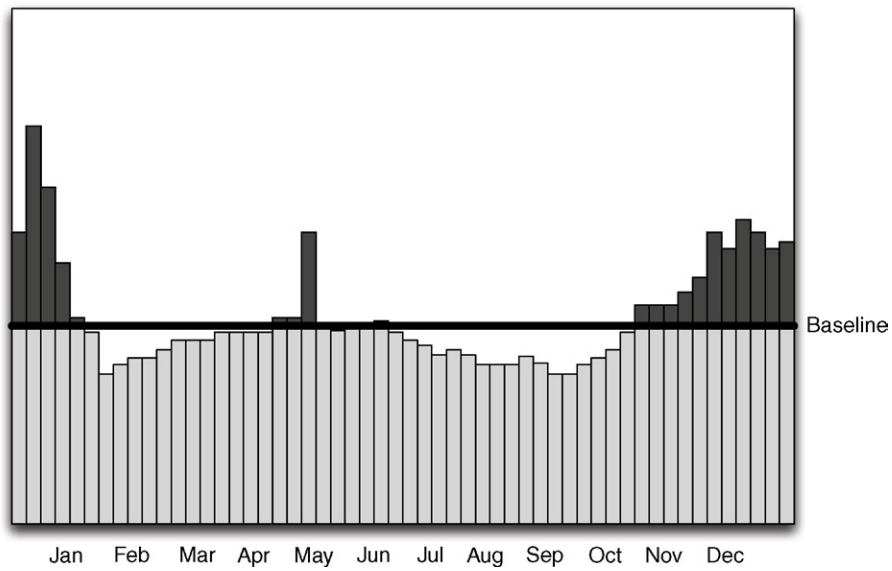

FIGURE 11.1 A flat average of all events over one year.

process to normal variations in process schedules. This concept is very similar to the statistical process control (SPC)/statistical quality control (SQC) $\bar{x}$ and R control chart comprising a control limit (equal to the baseline) with upper and lower control limits (UCL/LCL) that are used to signify events that are out of normal allowable tolerances.

This may or may not be useful for operations management; in a security context, this type of baseline provides little value. Knowing that 59,421,102 events over 30 days = 1,980,703 events per day average cannot tell us if the current day's event volume of 2,000,000 is meaningful or not, without some additional context. Does the yearly average include weekends and other periods of downtime? If it does, the actual per day expected values of a workday could be considerably higher. For purposes of behavioral analysis, a more applicable method would be a similar calculation that excludes known periods of downtime and creates a flat baseline that is more relevant to periods of operation. Better still are time-correlated baselines, where an observed period of activity is baselined against data samples taken over a series of similar time periods. That is, if looking at data for one (1) week, the baseline might indicate the expected patterns of behavior over a period of several weeks. Figure 11.2 illustrates how this affects the flatline average with a curved baseline that visualizes a drop in activity during weekends and shows an expected peak on Thursdays. Note that sufficient historical data are required to calculate time-correlated baselines.

Time-correlated baselines are very useful because they provide a statistical analysis of observed activity within relevant contexts of time—essentially providing historical context to baseline averages.[1] Without such a baseline, a spike in activity on Thursday might be seen as an anomaly and spur an extensive security analysis,

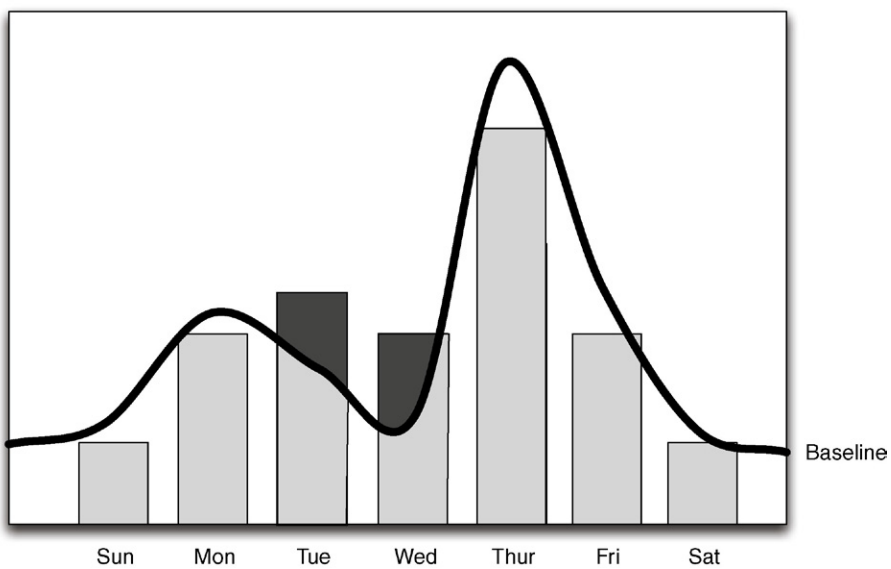

FIGURE 11.2 A time-correlated baseline shows dip on weekends, peak on Thursdays.

rather than being clearly indicated as normal behavior. Consider that there may be scheduled operations at the beginning of every month, at specific times of the day, or seasonally, all causing expected changes in event volumes.

Baselines, in whatever form, can be obtained in several ways, all beginning with the collection of relevant data over time, followed by statistical analysis of that data. Although statistical analysis of any metric can be performed manually, this function is often supported by the same product/system used to collect the metric, such as a Data Historian or an SIEM system (see Table 11.2 for examples).

Table 11.2 Measurement and Analysis of Baseline Metrics

Behavior	Measured Metric(s)	Measured by	Analyzed by
Network traffic	• Total unique Source IPs • Total unique Destination IPs • Total unique TCP/UPD ports • Traffic Volume (total flows) • Traffic Volume (total bytes) • Flow duration	• Network switch/ router flow logs (i.e. netFlow, jFlow, sFlow, or similar) • Network probe (i.e. IDS/IPS, network monitor, etc.)	• Network Behavior Anomaly Detection (NBAD) system • Log Management system • SIEM system
User activity	• Total unique active users • Total logons • Total logoffs • Logons by user • Logoffs by user • Activity (e.g. configuration changes) by user NOTE: user activity may need additional layers of correlation to consolidate multiple usernames/accounts associated with a single user	• Application Logs • Database logs and/or transaction analysis • Application logs and/or session analysis • Centralized authentication (LDAP, Active Directory, IAM)	• Log Management system • SIEM system
Process/ control behavior	• Total unique function codes • Total number per individual function code • Total set point or other configuration changes	• Industrial Protocol Monitor • Application Monitor • Data Historian tags	• Data Historian • SIEM System
Event/ incident activity	• Total events • Total events by criticality/ severity • Total events by security device	• Security device (i.e. firewall, IPS) logs	• Application Monitor • Industrial Protocol Filter

ANOMALY DETECTION

An anomaly is simply something that happens outside of normal defined parameters or boundaries of operation. Many firewalls and IDS/IPS devices may support anomaly detection directly, providing an additional detection capability at the conduits existing at a zone's perimeter. Holistically, all behaviors can be assessed for more systematic anomalies indicative of larger threats. Luckily, anomalies could be easily identified having defined expected (baseline) behaviors. In addition, many automated systems—including NBAD, log management, and SIEM systems—are available to facilitate anomaly detection across a number of different sources.

Behavioral anomaly detection is useful because there is no dependency upon a detection signature, and therefore unknown threats or attacks that may utilize zero-day capabilities can be identified. In addition, although often thought of exclusively in terms of network anomalies, any metric that is collected over time can be statistically analyzed and used for anomaly detection.

For example, an unexpected increase in network latency—measurable by easily obtained network metrics, such as TCP errors, the size of the TCP receive window, the round-trip duration of a ping—can indicate risk to the industrial network.[2] However, as can be seen in Table 11.3, anomalies can indicate normal, benign variations in behavior as well as potential threats. In other words, the rate of false positives tends to be higher using anomaly detection techniques.

Table 11.3 Examples of Suspicious Anomalies

Normal Behavior	Anomaly	Detected By	Indication
All Modbus communications to a group of PLCs originates from the same three HMI workstations	A fourth system communicates to the PLCs	• A >20% increase in the number of unique source IP addresses, from analysis of: Network flows • Security event logs from firewalls, IPS devices, etc. • Application logs • Etc.	• A new, unauthorized device has been plugged into the network (e.g. an administrator's laptop) • A rogue HMI is running using a spoofed IP address • A new system was installed and brought online
Every device has a single MAC address and a single IP address	An IP address is seen originating from two or more distinct MAC addresses	• >1 MAC Addresses per IP, from analysis of: Network flows • Security event logs from firewalls, IPS devices, etc. • Application logs • Etc.	• An attacker is spoofing an IP address • A device has failed and been replaced with new hardware

Table 11.3 Examples of Suspicious Anomalies *(cont.)*

Normal Behavior	Anomaly	Detected By	Indication
Process within a Control System zone is running for extended periods	Traffic increases above expected volumes	A >20% increase in the total network traffic, in bytes, from analysis of network flows	• An unauthorized service is running • A network scan or **penetration test** is being run • A shift change is underway • A new batch or process has started
Traffic decreases below expected levels	A >20% decrease in the total network traffic, in bytes, from analysis of network flows	• A service has stopped running • A networked device has failed or is offline • A batch or process has completed	
Changes to Controller Logic within BPCS, SIS, PLC, RTU	Industrial network monitor such as a SCADA IDS Ladder Logic/Code Review	• Any variation in the individual function codes and/or frequency of any function code, from analysis of Industrial Protocol Monitors • Application Monitors • SCADA IDS/IPS logs	• A process has been altered • A new process has been implemented • An old process has been removed • A process has been sabotaged
Authorized Users log on to common systems at the beginning of a shift	• Unauthorized user logs on to a system normally accessed by administrators only • Authorized users log on to a system outside of normal shift hours • Authorized users log on to unknown of unexpected systems	• Any variation seen from analysis of authentication logs from Active Directory Operating System logs • ICS Application Logs	• Personnel changes have been made • An administrator is on leave or absent and duties have been delegated to another user • A rogue user has authenticated to the system • An administrator account has been compromised and is in use by an attacker

Analyzing IT vs. OT Metrics

Up to this point, the discussion of anomaly detection has focused largely on security events derived from information technology (IT) tools. Even when looking at specialized security products for industrial network monitoring, these devices operate on the same paradigm as IT security devices to detect and block suspicious and/or "out of policy" events, subsequently generating an alert.

Anomaly Detection Tools

Anomaly detection can be done using anything from "gut feelings," to manual statistical analysis using a spreadsheet or mathematical application, to specialized statistics software systems, to network and security data analysis systems, such as certain log management and SIEM systems. Time-series databases, such as those used by Data Historians, can also be used for anomaly detection. While these systems do not typically represent anomalies within the specific context of network security, a Historian configured to show comparative overlays of security events over time could easily identify dangerous anomalies that might indicate a cyber-attack.

NBAD, log management, and SIEM tools are predominantly used for security-related anomaly detection. NBAD systems are focused exclusively on network activity and may or may not support the specific industrial network protocols used within an ICS environment. As such, the use of a log management or SIEM system may be better suited for anomaly detection in industrial networks. For example, Figure 11.3 shows a visual representation of anomalous authentication behavior for the administrative user (on the right) versus the same data shown without context (on the left); the security tool has done the necessary statistical analysis to show a 184% increase in administrator logins and has also brought that anomaly to the attention of the security analyst.

As shown in Table 11.3, this requires that the log management or SIEM system is used to collect relevant data over time from those systems used in perimeter and interior zone security, as well as any relevant network traffic data obtained from network switches and routers.

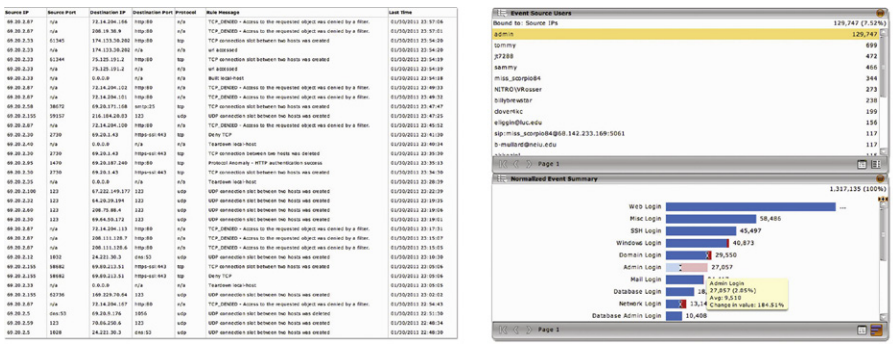

FIGURE 11.3 Representation of anomalous administrator logins using a SIEM system.

TIP

When selecting an analysis tool for industrial network anomaly detection, consider the greatest relevant time frame for analysis and ensure that the system is capable of automating anomaly detection over sufficient periods of time. Many systems, such as log management and SIEM systems, are not designed exclusively for anomaly detection and may have limitations as to how much information can be assessed and/or for how long.

To ensure the tool is right for the job, look at the operational lifespan of specific processes and use time-correlated baselines to determine normal activities for those processes. If a process takes 3 h, analysis of $n \times 3$ h of process data is needed for anomaly detection, where n represents the number of sampled operations. The greater the n, the more accurate the baseline and associated anomaly detection.

TIP

There are ICS network monitoring and intrusion detection systems available that automatically model normal and acceptable network behavior, and generate alerts whenever some network devices perform activities that diverge from their intended operation. For adequate behavior-based detection, these systems should first analyze network communications and generate a behavioral baseline—a valuable blueprint that defines communication patterns, protocols, message types, message fields, and field values that are normal for the monitored process. A review of the "blueprint" can reveal network and system misconfigurations (e.g. rogue devices), unintended communications, and unusual field values employed in the network. Continuous monitoring is then able to detect whenever network devices perform unintended activities—or anomalies outside the normal band.

This type of continuous monitoring is also useful for reporting observed network communications—in terms of communication patterns, protocols, and protocol message types normally used by the devices in the network—to additional security analytics tools, such as SIEM or anomaly behavior analysis systems, which are then able to perform even deeper analysis over longer periods of time.

BEHAVIORAL WHITELISTING

Whitelisting is well understood in the context of access control and application whitelisting (AWL) for host malware prevention. However, the concept of whitelisting has many roles within control system environments, where access, communication, processes, policies, and operations are all well-defined. Using the controlled nature of these systems and the zone-based policies defined in Chapter 9, "Establishing Zones and Conduits," whitelists can be defined for a variety of network and security metrics, including users, assets, applications, and others.

Whitelists can be actively enforced via a Deny !Whitelist policy on a firewall or IPS, or can be used throughout a network by combining network-wide monitoring and exception reporting with dynamic security controls. For example, if an exception is seen to a policy within a zone, a script can be run to tighten the specific perimeter defenses of that zone at all affected conduits.

USER WHITELISTS

Understanding user activity—especially of administrative users—is extremely useful for detecting cyber-attacks, both by insiders (e.g. intentional actors like a disgruntled employee, or unintentional actors like the control system engineer or subcontractor/vendor) as well as by outside attackers. Locking critical functions to administrative personnel, and then following best practices of user authentication and access control, means that an attack against a critical system should have to originate from an administrative user account. In reality, enumeration is a standard process in a cyber-attack because administrative accounts can be used for malicious intent (see Chapter 8, "Risk and Vulnerability Assessment"). They can be hijacked or used to escalate other rogue accounts in order to enable nonauthorized users' administrator rights.

NOTE

It should be pointed out that the term "administrator" does not have to mean a Windows Administrator account, but could represent a special Windows Group or Organizational Unit that has been established containing users with "elevated" privileges for particular applications. Some ICS vendors have implemented this concept, and facilitate the creation of separate application administrative roles from Windows administrative roles.

NOTE

Many ICS applications were developed and commissioned when cyber security was not a priority. The applications may require administrative rights to execute properly, and may even require execution from an administrator interactive account. These represent a unique problem discussed not only earlier, but also in Chapter 7, "Hacking Industrial Systems" due to the fact that if these applications or services can be exploited, the access level of the resulting payload is typically at the same level as the compromised component—the administrator in this case!

TIP

It is important to understand the ICS application software that is installed within a given facility, not only in terms of potential vulnerabilities within the application code base, but also implementation or configuration weaknesses that can easily be exploited. It is typically not possible for a user to assess the software coding practices of their ICS vendor. The US Department of Homeland Security (DHS) has developed the "Cyber Security Procurement Language for Industrial Control Systems"[3] guidance document that provides useful text that can be added to technical specifications and purchasing documents to expose and understand many hidden or latent potential weaknesses within the ICS components.

Fortunately, authorized users have been identified and documented (see Chapter 9, "Establishing Zones and Conduits"), and this allows us to whitelist user activities. As with any whitelist, the list of known users needs to be established and then compared to monitored activity. Authorized users can then be identified using a

directory service or an Identity and Access Management (IAM) system, such as Lightweight Directory Access Protocol (LDAP) included with Microsoft Active Directory, or other commercial IAM systems from IBM, Oracle, Sun, and others.

As with exception reporting, the whitelist is first defined and then monitored activity is compared against it. If there is an exception, it becomes a clear indicator that something outside of established policies is occurring. All known good user accounts are used as a detection filter against all login activity in the case of a user whitelist. If the user is on the list, nothing happens. If the user is not on the list, it is assumed bad and an alert is sent to security personnel. This accomplishes an immediate flag of all rogue accounts, default accounts, or other violations of the authentication policies. In early 2011, a security researcher was able to uncover hard-coded credentials within a PLC, and then used these credentials to gain shell access to the PLC.[4]

NOTE

In the case of hidden accounts and other hard-coded backdoor authentications, normal connections would also be flagged as an exception, because those accounts would most likely not appear on the whitelist. This could generate a potential excess of false-positive alerts. However, it would also draw attention to the existence of accounts that leverage default authentication within the system so that these accounts could be more closely monitored. For example, the WinCC authentication (used as one propagation mechanism in the Stuxnet campaign) could be monitored in conjunction with baseline analysis. If the default account was then used by new malware that was developed with knowledge learned from Stuxnet, it would still be possible to detect the threat via anomaly detection.

ASSET WHITELISTS

Once an inventory of cyber assets is completed—either automatically via an appropriate soft and "friendly" network scan (see Chapter 8, "Risk and Vulnerability Assessment") or manual inventory—the resulting list of known, authorized devices can be used to whitelist known good network devices.

Unlike perimeter-based security policies that may only allow known good devices into a zone or "inter-zone," a network asset whitelist can be applied to devices within a zone or "intra-zone." If a spoofed address or rogue device appears within a zone, it can still be detected via exception reporting against the list of known good devices so that action can be taken.

A classic use case for asset whitelisting is the use of "sneaker net," which can be used to carry files (documents, databases, applications) past perimeter defenses and attached directly to a protected network, well within a secure zone. This could be benign—an employee bringing a smart phone inside a control system that has Wi-Fi enabled—or it could be a deliberate vehicle for sabotage. Either way, the IP address of the device will be detected by switches, routers, network monitors, and security devices, and will eventually be seen in logs or events that are centralized and managed, as illustrated in Figure 11.4. At this point, simple comparison against the defined whitelist will identify the presence of an unauthorized device. This example represents significant risk, as the mobile device (smart phone in this case) also

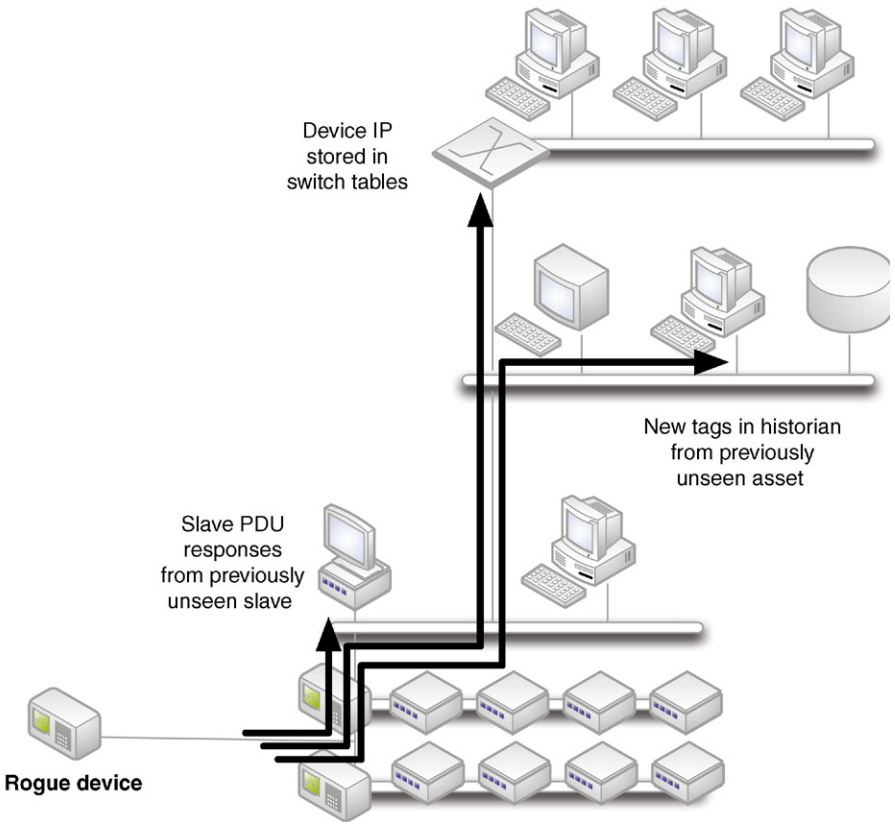

FIGURE 11.4 Information flow relevant to a rogue device IP.

connects directly to a 3G or 4G cellular network, which bypasses all defensive measures of the electronic security perimeter, and opens the zone up for attack or further exploitation.

TIP

One easy and effective method to prevent the introduction of unauthorized or foreign devices in a secure ICS zone is by disabling dynamic hardware addresses (e.g. media access control address) on the network switches within the zone. Default switch configurations allow dynamic creation of MAC tables within the switch effectively allowing any newly discovered device to begin forwarding and receive traffic. Disabling this feature not only secures the zone from intentional and malicious actors, but also from unintentional insiders accidently connecting devices not authorized for use within the zone—as defined by the security goals of the zone (see Chapter 9, "Establishing Zones and Conduits").

The whitelists themselves would need to be generated and applied to the central management system—most likely a log management or SIEM system that is capable of looking at device metrics across the entire network. Depending upon the specific

monitoring product used, the whitelist might be built through the use of a defined system variable (much like the generation of zone-specific variables in firewalls and IDS/IPS devices, as discussed in Chapter 10, "Implementing Security and Access Controls"), configurable data dictionaries, manually scripted detection signatures, and so on.

APPLICATION BEHAVIOR WHITELISTS

Applications themselves can be whitelisted per host using an AWL product. It is also possible for the application behavior to be whitelisted within the network. As with asset whitelisting, application behavior whitelists need to be defined so that good behavior can be differentiated from bad behavior. A central monitoring and management system can utilize application behavior whitelists by defining a variable of some sort within a log management or SIEM system just like asset whitelists. However, because of the nature of industrial network protocols, many application behaviors can be determined directly by monitoring those protocols and decoding them in order to determine the underlying function codes and commands being executed (see Chapter 6, "Industrial Network Protocols"). This allows for in-line whitelisting of industrial application behavior in addition to network-wide whitelisting offered by a log management or SIEM system. If in-line whitelisting is used via an industrial security appliance or application monitor, network whitelisting may still be beneficial for assessing application behavior outside of industrial control systems (i.e. for enterprise applications and ICS applications that do not utilize industrial protocols).

Some examples of application behavior whitelisting in industrial networks include

- Only read-only function codes are allowed.
- Master Protocol Data Units (PDU) or Datagrams are only allowed from predefined assets.
- Only specifically defined function codes are allowed.

Some examples of application behavior whitelisting in enterprise networks include

- Only encrypted HTTP web traffic is allowed and only on Port 443.
- Only POST commands are allowed for web form submissions.
- Human–machine interface (HMI) applications are only allowed on predefined hosts.

Some examples of application behavior whitelisting across both environments together include

- Write commands are only allowed in certain zones, between certain assets, or even during certain times of the day.
- HMI applications in supervisor networks are only allowed to use read functions over authorized protocols.

In other words, unlike AWL systems that only allow certain authorized applications to execute, application behavior whitelisting only allows applications authorized to execute to function in specifically defined ways on the network.

For example, an AWL system is installed on a Windows-based HMI. The AWL allows for the HMI application to execute, as well as a minimal set of necessary operating system services, and the networking services required to open Modbus/TCP network sockets so that the HMI can communicate to a series of RTUs and PLCs. However, the AWL does not control how the HMI application is used, and what commands and controls it can enforce on those RTUs and PLCs. A disgruntled employee can shut down key systems, randomly change set points, or otherwise disrupt operations using an HMI even though it is protected by AWL. Network-based application behavior whitelisting looks at how the HMI application is being used and compares that to a defined whitelist of authorized commands—in this case, a list of known good Modbus function codes. Functions that are not explicitly defined may then be actively blocked or they may be allowed but the system may generate an alert to notify administrators of the violated policy.

Industrial protocol or application monitoring tools should possess a base understanding of industrial protocols and their functions, allowing behavioral whitelists to be generated directly within the device. For network-wide behavioral whitelisting, variables or data dictionaries need to be defined. Common variables useful in application behavioral whitelisting include these same application function codes—the specific commands used by industrial protocols, ideally organized into clear categories (read, write, system commands, synchronization, etc.).

NOTE

It has probably become clear that there is a great deal of similarity between application behavior whitelisting at the host-level and deep-packet inspection at the network-level. Both technologies require application and/or protocol knowledge, and both provide a mechanism for an additional layer of protection beyond what or who is allowed to execute commands to what commands can be executed. These technologies should be appropriately deployed based on the target security level desired within a particular zone.

Examples of Beneficial Whitelists

Many whitelists can be derived using the functional groups defined in Chapter 9, "Establishing Zones and Conduits." Table 11.4 identifies some common whitelists, and how those whitelists can be implemented and enforced.

Smart-Lists

The term "Smart-Lists" was first introduced at the SANS Institute's 2010 European SCADA and Process Control Summit in London, United Kingdom. "**Smart-Listing**" combines the concept of behavioral whitelisting with a degree of deductive intelligence. Where blacklists block what is known to be bad, and whitelists only allow what is known to be good, Smart-Lists use the latter to help dynamically define the former.

Table 11.4 Examples of Behavioral Whitelists

Whitelist	Built Using	Enforced Using	Indications of a Violation
Authorized devices by IP	• Network monitor or probe (such as a Network IDS) • Network scan	• Firewall • Network Monitor • Network IDS/IPS	A rogue device is in use
Authorized applications by port	• Vulnerability assessment results • Local service scan • Port scan	• Firewall • Network IDS/IPS • Application Flow Monitor	A rogue application is in use
Authorized applications by content	• Application Monitor	An application is being used outside of policy	
Authorized Function Codes/Commands	• Industrial network monitor, such as an ICS IDS • Ladder Logic/ Code Review	• Application Monitor • Industrial Protocol Monitor	A process is being manipulated outside of policy
Authorized Users	• Active Directory Services • IAM	• Access Control • Application Log Analysis • Application Monitoring	A rogue account is in use

For example, if a critical asset is using AWL to prevent malicious code execution, the AWL software will generate an alert when an unauthorized application attempts to execute. What can now be determined is that the application is not a known good application for that particular asset. However, it could be a valid application that is in use elsewhere, and has attempted to access this asset unintentionally. A quick correlation against other whitelists can then determine if the application under scrutiny is an acceptable application on other known assets. If it is, the "Smart-Listing" process might result in an informational alert and nothing more. However, if the application under scrutiny is not defined anywhere within the system as a known good application, the Smart-Listing process can deduce that it is malicious in nature. It then defines it within the system as a known bad application and proactively defends against it by initiating a script or other active remediation mechanism to block that application wherever it might be detected.

"Smart-Listing" therefore combines what we know from established whitelists with deductive logic in order to dynamically adapt our blacklist security mechanisms (such as firewalls and IPS devices) to proactively block newly occurring threats. This process is illustrated in Figure 11.5. First, an alert is generated that identifies a violation of an established policy. Next, the nature of that alert is checked against other

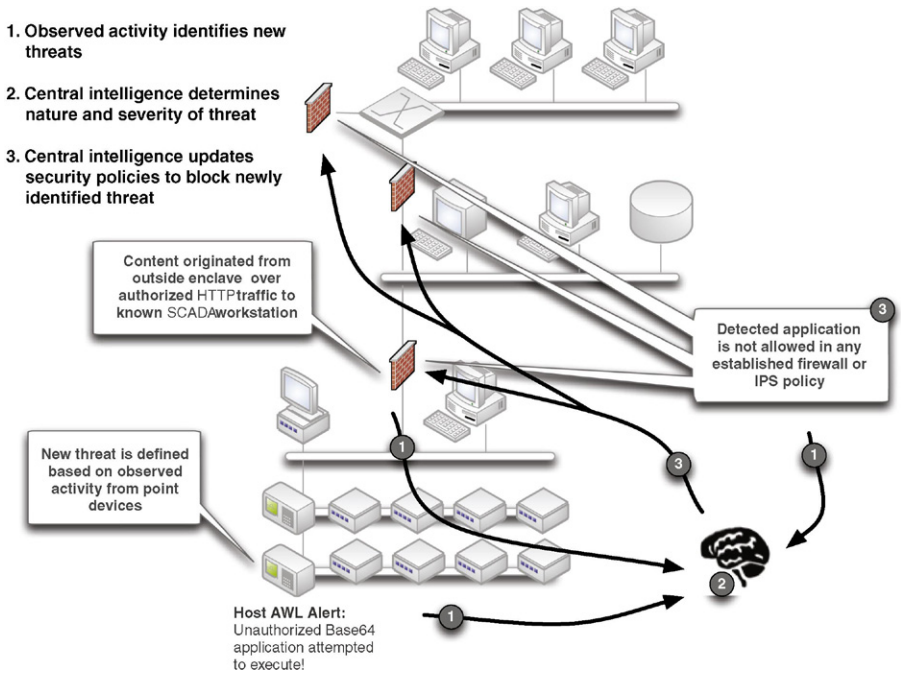

FIGURE 11.5 Smart-listing.

system-wide behavior. Finally, a decision is made—if it is "bad" a script or other automation service may be used to dynamically update firewall, IDS/IPS, and other defenses so that they can actively block this activity. If not, the activity might generate an alert, or be ignored.

Smart-Listing is a relatively new concept that could greatly benefit zone defenses by allowing them to automatically adapt to evasive attacks as well as insider attacks. Smart-Listing is especially compelling when used with overarching security management tools (see Chapter 12, "Security Monitoring of Industrial Control Systems"), as it requires complex event association and correlation. Although it has yet to be determined how widely security analysis and information management vendors will adopt this technique and whether ICS suppliers will endorse this approach, at present the techniques can be performed manually, using any number of log management or SIEM tools.

THREAT DETECTION

Used independently, the specific detection techniques discussed up to this point—security device and application logs, network connections, specific alerts generated by exception reporting or anomaly detection, and violations of whitelists—provide

valuable data points indicating events where a specific policy was violated. Even simple attacks consist of multiple steps. For the detection of an incident (vs. a discrete event), it is necessary to look at multiple events together and search for broader patterns. For example, many attacks will begin with some form of assessment of the target, followed by an enumeration technique, followed by an attempt to successfully authenticate against an enumerated account. (The remaining steps of elevating local privileges, creating persistent access, and covering tracks leave easy indicators for the numerous security controls described to this point.) This pattern might equate to firewall alerts indicating a ping sweep, followed next by access to the `sam` and `system` files, ending with a brute force login. The detection of this larger threat pattern is known as event correlation. As cyber-attacks continue to increase in sophistication, event correlation methods have continued to expand. They consider event data from a wider network of point security devices, additional event contexts, such as user privileges or asset vulnerabilities, and search for more complex patterns.

In looking at Stuxnet, another factor was introduced that further complicated the event correlation process. Prior to Stuxnet, a threat had never before involved events from both IT and OT systems. The correlation of events across both IT and OT systems is also necessary with the evolution of threat patterns that traverse both domains. The problem is that event correlation systems were not designed to accommodate OT systems, presenting challenges in the detection of the most serious threats to industrial networks.

EVENT CORRELATION

Event correlation simplifies the threat detection process by making sense of the massive amounts of discrete event data, analyzing it as a whole to find the important patterns and incidents that require immediate attention. Although early event correlation focused on the reduction of event volumes in order to simplify event management—often through filtering, compressing, or generalizing events[5]—newer techniques involve state logic to analyze event streams as they occur, performing pattern recognition to find indications of network issues, failures, attacks, intrusions, and so on.[6] Event correlation is useful in several ways, including facilitating human security assessments by making the large volumes of event data from a wide variety of sources more suitable for human consumption and comprehension, by automatically detecting clear indications of known threat patterns to easily detect incidents of cyber-attack and sabotage, and by facilitating the human detection of unknown threat patterns through event normalization. The process of event correlation is depicted in Figure 11.6.

Events are first compared against a defined set of known threat patterns or "correlation rules." If there is a match, an entry is made in a (typically) memory-resident state tree; if another sequence in the pattern is seen, the rule progresses until a complete match is determined. For example, if a log matches the first condition of a rule, a new entry is made in the state tree, indicating that the first condition of a rule has been met. As more logs are assessed, there may be a match for a subsequent condition

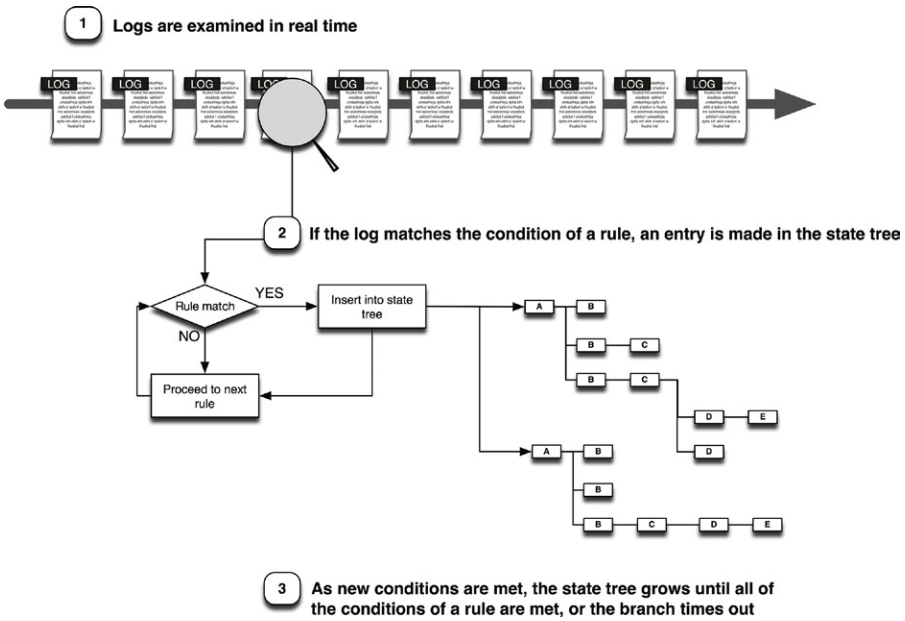

FIGURE 11.6 The event correlation process.

of an existing branch at which point that branch is extended. A log may meet more than one condition of more than one rule, creating large and complex state trees. For example, even a simple "brute force attack" rule can create several unique branches. Consider the rule

```
If [5 consecutive failed logins] from [the same source IP] to [the
same destination IP] within [5 minutes]
```

This example would create one branch for the first failed login event "A" from any IP address to any other IP address. The next matching login event "B" would extend that initial branch while also generating a new branch (with a new timer):

```
A + B
B
```

The third matching login event "C" would extend the first two branches while also creating a third:

```
A + B + C
B + C
C
```

This will continue *ad infinitum* until all of the conditions are met, or until a branch's timer expires. If a branch completes (i.e. all conditions are met), the rule triggers.

Note that events are collected from many types of information sources, such as firewalls, switches, and authentication services. They must be normalized into a common event taxonomy before they can be effectively correlated. Normalization categorizes activities into a common framework so that similar events can be correlated together even if the originating log or event formats differ.[7] Without normalization, many additional correlation rules would be required in order to check a condition (in this example a failed login) against all possible variations of that event that may be present (Windows logins, Application logins, etc.).

For purposes of threat detection, the entire event correlation process is typically performed in memory at the time the individual logs and events are collected. Correlation can also be performed manually by querying larger stores of already collected events to find similar patterns.[8]

Examples of event correlation rules are provided in Table 11.5. Event correlation may be very basic (e.g. a brute force attack) or highly complex—up to and including tiered correlation that consists of correlation rules within correlation rules (e.g. a brute force attack followed by a malware event).

Data Enrichment

Data enrichment refers to the process of appending or otherwise enhancing collected data with relevant context obtained from additional sources. For example, if a username is found within an application log, that username can be referenced against a central IAM system (or ICS application if Application Security is deployed) to obtain

Table 11.5 Example Event Correlation Rules

Threat Pattern	Description	Rule
Brute force attack	Passwords are guessed randomly in quick succession in order to crack the password of a known user account	A number N of Failed Logon events, followed by one or more Successful Logon events, from the same Source IP
Outbound Spambot behavior	A spambot (malware designed to send spam from the infected computer) is sending bulk unsolicited e-mails to outside addresses	A large number N of Outbound SMTP events, from one internal IP Address, each destined to a unique e-mail address
HTTP command and control	A hidden (covert) communication channel inside of HTTP (overt) is used as a command and control channel for malware	HTTP traffic is originating from servers that are not HTTP servers
Covert botnet, command, and control	A distributed network of malware establishing covert communications channels over applications that are otherwise allowed by firewall or IPS policy	Traffic originating from N number of $ControlSystem_Zone01_Devices to !$ControlSystem_Zone01_Devices with contents containing Base64 coding.

the user's actual name, departmental roles, privileges, and so on. This additional information "enriches" the original log with this context. Similarly, an IP address can be used to enrich a log file, referencing IP reputation servers for external addresses to see if there is known threat activity associated with that IP address, or by referencing geolocation services to determine the physical location of the IP address by country, state, or postal code (see "Additional Context" in Chapter 12, "Security Monitoring of Industrial Control Systems," for more examples of contextual information).

CAUTION

Many of the advanced security controls described in this chapter leverage the use of external threat intelligence data. It is always important to remember to follow strict security policies on network connectivity between trusted control zones and less-trusted enterprise and public (i.e. Internet) zones. This can be addressed by proper location of local assets requiring remote information, including the creation of dedicated "security zones" within the semitrusted DMZ framework.

Data enrichment can occur in two primary ways. The first is by performing a lookup at the time of collection and appending the contextual information into the log. Another method is to perform a lookup at the time the event is scrutinized by the SIEM or log management system. Although both provide the relevant context, each has advantages and disadvantages. Appending the data at the time of collection provides the most accurate representation of context and prevents misrepresentations that may occur as the network environment changes. For example, if IP addresses are provided via the Dynamic Host Configuration Protocol (DHCP), the IP associated with a specific log could be different at the time of collection than at the time of analysis. Although more accurate, this type of enrichment also burdens the analysis platform by increasing the amount of stored information. It is important to ensure that the original log file is maintained for compliance purposes, requiring the system to replicate the original raw log records prior to enrichment.

The alternative, providing the context at the time of analysis, removes these additional requirements at the cost of accuracy. Although there is no hard rule indicating how a particular product enriches the data that it collects, traditional Log Management platforms tend toward analytical enrichment, whereas SIEM platforms tend toward enrichment at the time of collection, possibly because most SIEM platforms already replicate log data for parsing and analysis, minimizing the additional burden associated with this type of enrichment.

Normalization

Event normalization is a classification system that categorizes events according to a defined taxonomy, such as the Common Event Expression Framework provided by the MITRE Corporation.[9] Normalization is a necessary step in the correlation process, due to the lack of a common log format.[10] Table 11.6 provides a comparison of authentication logs associated with logon activity from a variety of sources.

NOTE

In 2006, security software company ArcSight (purchased by Hewlett-Packard in 2010), saw the need to improve the interoperability of devices in terms of how event data are logged and transmitted. The problem at the time was that each vendor had their own unique format for reporting event information that was often found to lack the necessary information needed to integrate these events with other systems. This new format was called the Common Event Format (CEF) and defined a syntax for audit log records comprised of a standard header and a variable expression formatted as key-value pairs. CEF allows vendors of both security and non-security devices to structure their syslog event data making it more easily parsed.[11]

Although each example in Table 11.6 is a logon, the way the message is depicted varies sufficiently such that without a compensating measure, such as event normalization, a correlation rule looking for "logons" would need to explicitly define each known logon format. In contrast, event normalization provides the necessary categorization so that a rule can reference a "logon" and then successfully match an event against any variety of logons. Most normalization taxonomies utilize a tiered categorization structure because this level of generalization may be too broad for the detection of specific threat patterns, as illustrated in Figure 11.7.

Cross-Source Correlation

Cross-source correlation refers to the ability to extend correlation across multiple sources so that common events from disparate systems (such as a firewall and an

Table 11.6 Common Logon Events Depicted by Varying Log Formats[a]

Log Source	Log Contents	Description
Juniper firewall	<18> Dec 17 15:45:57 10.14.93.7 ns5xp: NetScreen device_id 5 ns5xp system-warning-00515: Admin User jdoe has logged on via Telnet from 10.14.98.55:39073 (2002-12-17 15:50:53)	Successful Logon
Cisco router	<57> Dec 25 00:04:32:%SEC_LOGIN-5-LOGIN_SUCCESS:Login Success [user:jdoe] [Source:10.4.2.11] [localport:23] at 20:55:40 UTC Fri Feb 28 2006	Successful Logon
Redhat Linux	<122> Mar 4 09:23:15 localhost sshd[27577]: Accepted password for jdoe from ::ffff:192.168.138.35 port 2895 ssh2	Successful Logon
Windows	<13> Fri Mar 17 14:29:38 2006 680 Security SYSTEM User Failure Audit ENTERPRISE Account Logon Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon account: JDOE Source Workstation: ENTERPRISE Error Code: 0xC000006A 4574	Successful Logon

[a]*A. Chuvakin, Content aware SIEM. http://www.sans.org/security-resources/idfaq/vlan.php, February, 2000 (cited: January 19, 2011).*

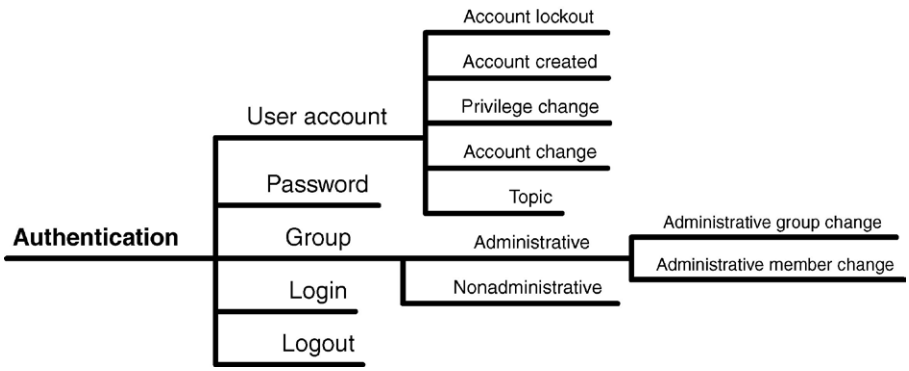

FIGURE 11.7 A partial representation of a tiered normalization taxonomy.

IPS) may be normalized and correlated together. As correlation systems continue to mature, the availability of single-source correlation is dwindling. Cross-source correlation remains an important consideration of threat detection capability. The more types of information that can be correlated, the more effective the threat detection will be, and the fewer false positives, as shown in Table 11.7.

As more systems are monitored (see Chapter 12, "Security Monitoring of Industrial Control Systems"), the potential for expanding cross-source correlation increases accordingly—ideally with all monitored information being normalized and correlated together.

Tiered Correlation

Tiered correlation is simply the use of one correlation rule within another correlation rule. For example, a brute force attempt on its own may or may not be indicative of a cyber incident. If it is a cyber-attack, there is no further determination of

Table 11.7 Single-Source vs. Cross-Source Correlation

Single-Source Correlation Example	Cross-Source Correlation Example
Multiple failed logon followed by one or more Successful logon	Multiple failed logon events by an Admin user of Critical Assets, followed by one or more Successful Logon
Any successful logon to a Critical Asset	Any Successful Logon to a Critical Asset, by either a Terminated Employee or by an Admin User at a time outside of Normal shift hours.
HTTP traffic is originating from servers that are not HTTP servers	HTTP traffic is originating from servers that are not HTTP servers' IP addresses with a geographic location outside of the United States

Table 11.8 Tiered Correlation Examples

Description	Rule
Brute force attack	A number *N* of Failed Logon events, followed by one or more Successful Logon events, from the same Source IP
Brute force malware injection	A number *N* of Failed Logon events, followed by one or more Successful Logon events, from the same Source IP, followed by a Malware Event
Brute force followed by internal propagation	A number *N* of Failed Logon events, followed by one or more Successful Logon events, from the same Source IP, followed by a Network Scan originating from the same Source IP
Internal brute force enumeration using known password	A number *N* of Failed Logon events from the same Source IP, each with a unique username but a different password

what the attack is, or its intent. By stacking correlation rules within other rules, additional rules can be enabled to target more specific attack scenarios, as shown in Table 11.8.

The third example in Table 11.8 illustrates the use of normalization within correlation by using a Malware Event as a general condition of the rule. The fourth example illustrates the value of content inspection for the purposes of threat detection by exposing application authentication parameters to the correlation engine.

CORRELATING BETWEEN IT AND OT SYSTEMS

Up until now, correlation has been discussed solely within the context of IT networks running standard enterprise systems and protocols. Operational Technology systems must also be analyzed, requiring that metrics within the OT network be correlated to events in the IT network. The challenge here is the disparity of the two system types, and the information collection models used within each. IT systems are monitored heavily for performance and security using a wide range of available tools, whereas OT systems are monitored primarily for process efficiency and performance using a more limited range of tools consisting of Data Historians, spreadsheets, and statistical modeling applications (see Chapter 12, "Security Monitoring of Industrial Control Systems").

Even benign network behaviors of the IT network can impact operations, and threats do exist across both IT and OT systems. By correlating IT conditions against OT conditions, a good deal can be determined about potential cyber incidents.[12] Table 11.9 shows and example of several instances where IT systems can impact OT systems.

To fully leverage the automated correlation capability built into most IT SIEM products, OT data must first be collected into the SIEM, and then the normalization of one metric to another must be made using a common threat taxonomy.

CAUTION

The ability to collect, interpret, and correlate data from disparate systems is vital to an effective security monitoring solution. The devices that comprise the network architectures must be able to communicate event data to a system that is equally capable of receiving these data. These concepts are progressive to OT networks, and is a primary reason why many ICS servers, workstations, and embedded devices do not support this capability. It is not uncommon for an ICS vendor to restrict additional components that can be installed on their assets in order to maintain not only continuous performance and availability to manufacturing operations, but also the long-term support required to service these systems for years to come. At the time of publishing, there are several companies offering "SCADA SIEM" or similar packages. As SCADA and ICS systems continue to incorporate more mainstream security features, the ability of commercial monitoring and analysis tools to support industrial systems will continue to improve. Many commercial security analysis systems lack the necessary context to understand the data being collected from industrial systems, limiting the value of their analytics. This trend will change as more security solution companies partner with ICS vendors in delivering integrated OT security solutions.

Table 11.9 Correlation of IT and OT Systems[a]

Incident	IT Event	OT Event	Condition
Network instability	Increased Latency, measured by TCP errors, reduction of TCP receive windows, increased round-trip TTL, etc.	Reduction in Efficiency, measured by historical batch comparisons	Manifestation of network condition in operational processes Deliberate cyber sabotage
Operational change	No detected event	Change to operational set points, or other process change(s)	Benign process adjustment Undetected cyber sabotage
Network breach	Detected threat or incident using event correlation, to determine successful penetration of IT system(s)	Change to operational set points, or other process change(s)	Benign process adjustment Undetected cyber sabotage
Targeted incident	Detected threat or incident directly targeting industrial SCADA or DCS systems connected to IT networks	Abnormal change to operational set points, unexpected PLC code writes, etc.	Potential "Stuxnet-class" cyber incident or sabotage

[a]*B. Singer, Correlating Risk Events and Process Trends. Proceedings of the SCADA Security Scientific Symposium (S4). Kenexis Security Corporation and Digital Bond Press, 2010.*

SUMMARY

A larger picture of security-related activity begins to form when zone security measures are in place. Measuring these activities and analyzing them can detect exceptions from the established security policies. In addition, anomalous activities can be identified so that they may be further investigated.

This requires well-defined policies and also requires that those policies be configured within an appropriate information analysis tool to ensure enforcement of those policies. Just as with perimeter defenses to a 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," which can help strengthen perimeter defenses through dynamic firewall configuration or IPS rule creation.

The event information can be further analyzed by event correlation systems as various threat detection techniques are used together to find larger and broader patterns that are more indicative of serious threats or incidents. Though widely used in IT network security, event correlation is now 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—all rely on a rich base of relevant security information. Where does this security information come from? Chapter 12, "Security Monitoring of Industrial Control Systems," discusses what to monitor, and how, in order to obtain the necessary baseline of data required achieving "situational awareness" and effectively securing an industrial network.

ENDNOTES

1. F. Salo, Anomaly Detection Systems: Context Sensitive Analytics. NitroSecurity, Inc. Portsmouth, NH, December 2009.
2. B. Singer, Correlating Risk Events and Process Trends. Proceedings of the SCADA Security Scientific Symposium (S4). Kenexis Security Corporation and Digital Bond Press, Sunrise, FL, 2010.
3. U.S. Dept. of Homeland Security, "Cyber Security Procurement Language for Industrial Control Systems," September 2009.
4. D. Beresford, "Exploiting Siemens Simatic S7 PLCs," July 8, 2011. Prepared for Black Hat USA 2011.
5. R. Kay, QuickStudy: event correlation. Computerworld.com <http://www.computerworld.com/s/article/83396/Event_Correlation?taxonomyId=016>, July 28, 2003 (cited: February 13, 2011).
6. Softpanorama, Event correlation technologies. <http://www.softpanorama.org/Admin/Event_correlation/>, January 10, 2002 (cited: February 13, 2011).
7. The MITRE Corporation, About CEE (common event expression). <http://cee.mitre.org/about.html>, May 27, 2010 (cited: February 13, 2011).

8. M. Leland, Zero-day correlation: building a taxonomy. NitroSecurity, Inc. <http://www.youtube.com/watch?v=Xtd0aXeLn1Y>, May 6, 2009 (cited: February 13, 2011).
9. The MITRE Corporation, About CEE (common event expression). <http://cee.mitre.org/about.html>, May 27, 2010 (cited: February 13, 2011).
10. A. Chuvakin, Content aware SIEM. <http://www.sans.org/security-resources/idfaq/vlan.php>, February 2000 (cited: January 19, 2011).
11. ArcSight, "Common Event Format," Revision 16, July 22, 2010
12. B. Singer, Correlating risk events and process trends. Proceedings of the SCADA Security Scientific Symposium (S4). Kenexis Security Corporation and Digital Bond Press, 2010, Sunrise, FL.