30 April 2010

Detection and Analysis of Threats to the Energy Sector: DATES

Final Technical Report

DOE Instrument Number: DE-FC26-07NT43314

SRI Project: P18201

Prepared for:
DOE-National Energy Technology Laboratory
3610 Collins Ferry Road
P.O. Box 880
Morgantown, WV 26507-0880
Attn: Diane Hooie, DOE Project Officer

Prepared by:
Alfonso Valdes, Principal Investigator
SRI International, Award Recipient
afonso.valdes@sri.com, (650)859-4976

Teaming Members:
Sandia National Laboratories
ArcSight, Inc.
Invensys Process Systems

DISCLAIMER

Table of Contents

Executive Summary

This report summarizes Detection and Analysis of Threats to the Energy Sector (DATES), a project sponsored by the United States Department of Energy and performed by a team led by SRI International, with collaboration from Sandia National Laboratories, ArcSight, Inc., and Invensys Process Systems.

DATES sought to advance the state of the practice in intrusion detection and situational awareness with respect to cyber attacks in energy systems. This was achieved through adaptation of detection algorithms for process systems as well as development of novel anomaly detection techniques suited for such systems into a detection suite. These detection components, together with third-party commercial security systems, were interfaced with the commercial Security Information Event Management (SIEM) solution from ArcSight. The efficacy of the integrated solution was demonstrated on two testbeds, one based on a Distributed Control System (DCS) from Invensys, and the other based on the Virtual Control System Environment (VCSE) from Sandia. These achievements advance the DOE Cybersecurity Roadmap [DOE2006] goals in the area of security monitoring.

The project ran from October 2007 until March 2010, with the final six months focused on experimentation. In the validation phase, team members from SRI and Sandia coupled the two test environments and carried out a number of distributed and cross-site attacks against various points in one or both testbeds. Alert messages from the distributed, heterogeneous detection components were correlated using the ArcSight SIEM platform, providing within-site and cross-site views of the attacks. In particular, the team demonstrated detection and visualization of network zone traversal and denial-of-service attacks. These capabilities were presented to the DistribuTech Conference and Exhibition in March 2010.

The project was hampered by interruption of funding due to continuing resolution issues and agreement on cost share for four months in 2008. This resulted in delays in finalizing agreements with commercial partners, and in particular the Invensys testbed was not installed until December 2008 (as opposed to the March 2008 plan).

The project resulted in a number of conference presentations and publications, and was well received when presented at industry forums. In spite of some interest on the part of the utility sector, we were unfortunately not able to engage a utility for a full-scale pilot deployment.

Introduction

As Industrial Control Systems (ICS) rely more and more on commercial off-the-shelf (COTS) digital technologies, and have network connectivity to corporate networks and other systems, they become increasingly vulnerable to cyber attacks. The digital workstations used in control centers inherit many of the vulnerabilities of conventional IT systems, but lag in security best practices for a variety of reasons specific to ICS. Field devices are increasingly sophisticated, connected via TCP/IP networking technology and featuring real-time operating systems and in some cases web-based configuration.

While displays in control centers provide extensive diagnostic and control capability of remote field assets from a process point of view, they may be effectively blind to security issues in field networks. Moreover, the control center workstations may themselves be attacked, either from inadequately secured connections to business networks or via portable devices that enter the ICS environment.

Perimeter defenses complemented by high-fidelity monitoring are essential components of a defense in-depth strategy. Correctly configured switches and firewalls, along with careful network segmentation, can provide valuable perimeter defense for ICS, including DCS and SCADA (supervisory control and data acquisition). Even with strong perimeter defenses, security monitoring is required to make the system owner aware of attack attempts, penetration or circumvention of the defenses, and insider misuse. The DATES monitoring solution complements perimeter defenses and provides the ICS security operator a significantly improved level of situational awareness.

Detection and Analysis of Threats to the Energy Sector (DATES) is a detection and security information/event management (SIEM) solution specifically tailored to protect ICS used in the energy sector. Features of the DATES monitoring platform include

- Multiple detection algorithms, including an ICS-aware Snort knowledge base, as well as SRI's components for stateful packet inspection, probabilistic/Bayesian analysis, and event threading.
- A unique model-based detection capability, including a communication pattern anomaly detection module, which leverages the unique traffic characteristics of ICS to facilitate detection of novel attacks such as zero-day exploits.
- Non-intrusive network monitoring design based on passive listening and employing a separate network interface for event reporting. This makes the monitoring appliance invisible to conventional network scans, and guarantees that the critical function of the ICS is not affected at all.

- DATES monitoring components that interface with the advanced market-leading ArcSight SIEM Platform, and can easily be adapted to communicate with other types of event-consuming components.

DATES may be flexibly deployed in an ICS, with multiple instances of the detection component monitoring different network segments in the field and in the control center, communicating events to the SIEM console.

This report summarizes the results of the DATES project, and is organized as follows. The next two sections discuss the design and implementation for the intrusion detection system (IDS) and Security Information Event Management (SIEM) for process control systems. The following section describes the SRI and Sandia test environments. We then describe a series of experiments on the instrumented testbeds, demonstrating situational awareness and cross-site correlation of attacks as they cross PCS zone boundaries and escalate in severity. This is followed with a discussion of our outreach activities, and then a report summary. The appendix includes DATES papers published in the proceedings of two conferences.

Tailoring Intrusion Detection for Process Control Systems

Because process control systems typically consist of enterprise COTS components (e.g., commercial database systems running on Microsoft Windows) and process-control-specific components, such as Remote Terminal Units (RTUs) using Modbus TCP, our intrusion monitoring approach employs a suite of complementary sensors for both enterprise and process-control-specific subsystems to provide good detection coverage.

For monitoring enterprise networks and COTS components, we use Snort equipped with rulesets for enterprise networks, commercial host-based security solutions such as the Symantec Endpoint Protection system [SEP] and McAfee AntiVirus Enterprise system [McAfee], and EMERALD network-based intrusion detection sensors for enterprise networks (e.g., Bayesian TCP sensor for monitoring availability for network services). For monitoring the process-control-specific subsystem, we employ a complementary suite of intrusion detection technologies, including the PCS-specific Snort rules developed by Digital Bond [DB2010] and the model-based intrusion detection (also called anomaly detection) components, called eModbus and eFlowmon, developed under this project for process control systems. All the network intrusion detection technologies are integrated into a network appliance to facilitate ease of use.

Before we present our intrusion detection system, we will describe the reference architecture for process control systems used in this project to highlight some of its characteristics.

Reference Architecture

We consider a corporate or enterprise network (possibly Internet-facing), with clients that access resources such as historian servers in a demilitarized zone (DMZ) between corporate and control zones. These historian servers are populated by field control processors (FCP) or front end processors (FEP) in the control zone, which issue control commands to and poll data from devices in field networks. The control network typically contains assets such as the human-machine interface (HMI) and other workstations, which run control system applications on conventional computational platforms. The field network devices directly monitor and control a physical process, such as refining, manufacturing, or electric power generation/transmission/distribution. The corporate network is untrusted from the point of view of the control and field networks.

The system can operate with loss of the DMZ servers, and in many cases even with temporary loss of the control network. In this case, the field network operates autonomously for a time or is brought to a safe shutdown by a logically orthogonal safety instrumented system (SIS). As such, the field network is considered highest priority, the control network high, the DMZ medium, and the corporate low.

The expected traffic is regular by comparison to traffic on enterprise networks. Clients in the corporate zone may access the DMZ only over the protocols allowed for the historian server. In practice, traffic such as Windows RPC is often present as well, and has known vulnerabilities. Also, defenders must be aware of vulnerability exploits over the allowed protocols, as well as the possibility of hijacked TCP connections. Thus, while suspicious traffic from one zone to another should always trigger an alarm, the absence of such traffic is no guarantee that the system is not being attacked, so that additional techniques such as deep packet inspection and asset health monitoring are essential for defense in depth.

The intrusion detection framework in DATES contains multiple detection algorithms, combining conventional attack signature detection, protocol analysis, and a Bayesian component adapted from our EMERALD system [EMERALD, Bayes]. These detection approaches are geared to detect attacks similar to the attacks seen in enterprise networks. Process control networks increasingly use commodity platforms such as Microsoft Windows workstations, routing and switching equipment logically similar to enterprise counterparts, and standard Internet protocols such as TCP/IP (sometimes encapsulating a legacy control protocol such as Modbus) and HTTP/HTTPS. In addition to using intrusion detection components designed for enterprise networks, we employ control-system-specific monitors that can perform in-depth analysis of ICS protocols (such as Modbus) and can leverage the special-purpose characteristics of process control systems to facilitate anomaly detection. We believe this combination of intrusion detection components can provide a comprehensive monitoring capability for ICS.

The figure below depicts a representative architecture of the defended system, instrumented with the detection and correlation solutions described in this report. The architecture is applicable to control systems in such sectors as electricity distribution supervisory control and data acquisition (SCADA) or an oil refinery DCS. The test environments implemented at SRI and Sandia may be considered specific implementations of this architecture.

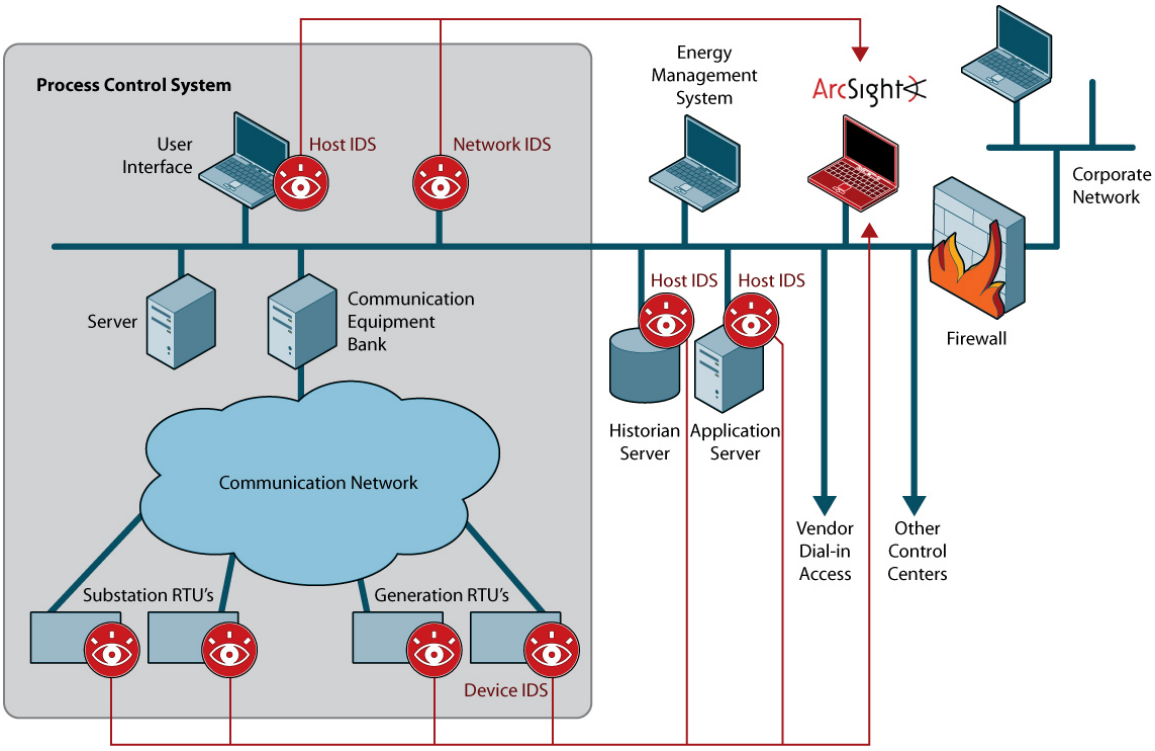


Figure 1. Representative architecture

Intrusion Detection for Enterprise Networks and Hosts

To monitor enterprise networks (such as the corporate network in the reference architecture) and COTS components (such as commercial database systems and operating systems), we employ a suite of best-of-the-breed intrusion detection sensors.

For network monitoring, we employ Snort equipped with selected rules developed by the intrusion detection community and more recent Snort rules from Emerging Threats [SNORT, Emerg], Bayesian sensors for detecting several important attack classes such as reconnaissance and asset distress [Bayes], and EMERALD eXpert sensors for performing deep packet inspection and stateful analysis for several key network protocols such as HTTP [EMERALD].

For host monitoring, we use commercial security solutions (in particular, SEP and McAfee) for monitoring machines running Microsoft Windows. These host-based security components may detect malicious activities that are not easily observable by network monitoring components, such as modification of security-critical files on the target hosts or attacks that are propagated via encrypted network connections, and thus can provide enhanced detection coverage.

Intrusion Detection for Process-control-specific Subsystems

Our control-system-specific monitoring solution employs both the signature-based approach and the anomaly detection approach. For the signature-based approach, we use Snort with the control-network-specific rules developed by Digital Bond, another performer in the National SCADA Text Bed (NSTB) program [DB2010]. These Snort rules detect attacks against Modbus TCP [Modbus] and DNP3 [DNP3], two important control system protocols. Digital Bond also provides signatures for ICCP, the Inter Control Center Protocol [ICCP]. ICCP is used in the multi-site experiment described below.

Anomaly Detection for Process-control-specific environments

Intrusion detection systems using anomaly detection (AD) techniques are not widely deployed in enterprise systems, because such systems typically exhibit highly variable behavior. As such, AD systems, particularly those based on learning normal system activity and alerting on abnormal activity, often alert on activity that is unusual, but not malicious, while failing to alert on malicious activity that recurs frequently enough to not appear unusual.

By contrast to enterprise systems, process control systems often exhibit regular and predictable communication patterns, which can be leveraged in an AD system. An attack launched against a process control network may exhibit communication patterns quite different from those observed during normal operations.

In our earlier work [S4], we demonstrated that these regularities can form the basis of a model-based IDS in control systems, where much of the expected behavior of the system can be coded into a fairly compact ruleset/model, which complements misuse detection rules used for detecting known malicious activities.

In this project, we extended this work in two directions. First, we developed Snort rules for characterizing the DNP3 protocol to detect deviations from the protocol specification. Second, because developing the models to specify the expected system behavior by hand is error-prone and time-consuming, we developed a learning-based communication pattern anomaly detection approach for process control systems.

In DATES, we extend the communication pattern AD technique to effectively learn normal flows and alert on statistical exceptions to the learned norms. We developed and experimented with two anomaly detection techniques: pattern-based anomaly detection for monitoring the patterns of hosts with which each host communicates, and flow-based anomaly detection for monitoring the traffic patterns for individual network flows. Moreover, we have implemented a flow-based anomaly detection sensor, called *eflowmon,* and integrated it with the EMERALD network intrusion detection appliance.

Our approach involves learning network communication patterns in process control networks by passively monitoring network traffic. Specifically, our IDS employs network flow information such as connection endpoints (i.e., source and destination IP addresses and port numbers), the rate of packet flow between network endpoints, and the set of hosts with which a host communicates. The IDS maintains a database of recent and historical network flow profiles observed in process control networks. A flow record is generated or updated as packets are observed. Detected network flow patterns are then evaluated against the learned historical norms. An observed pattern can either match an existing historical flow profile through reinforcement learning, or start a new pattern exemplar. The pattern exemplars are effectively different modes of observed activity, so our system does not require attack-free training data. The system alerts on observed flow patterns that are statistical exceptions to the learned norms. Specifically, we are interested in anomalies such as new network flows, significant changes in flow rates and packet length statistics, and the absence of expected network flows. These anomalies may correspond to network probing attacks, propagation of malware, introduction of rogue master or slave devices, flooding-based denial-of-service attacks, or attacks that cause host or service failure.

For the detailed description of our work on communication pattern anomaly detection, refer to our publication in the 2009 IEEE International Conference on Technologies for Homeland Security [HST 09].

Security Information and Event Management in Control Systems

Control systems monitor processes to collect process parameters and provide process alarms as two of their core functions. Process alarms are not necessarily indicative of malicious activity. To include an intrusion monitoring and situational awareness capability such as DATES risks burdening the operator with additional alarms, in this case from the intrusion detection framework. Correlation of alerts from intrusion detection systems is therefore essential in order to provide a succinct representation of potential cyber attacks against the system, including indications of severity and a capability for detailed drill-down.

Numerous SIEM systems have emerged in recent years to tackle the correlation challenge in enterprise networks. There have been some demonstrations of SIEM in the process control setting [LOGIIC]. The DATES SIEM capability extends this to comprehend a much greater variety in detection methods, a higher-fidelity description of the monitored environment, and the ability to visualize attacks. We have built our SIEM on the ArcSight system, which is a leading commercial SIEM solution.

Building Blocks for Correlation

Our alert correlation approach builds on several basic concepts, including incident classification, network zones, and asset types. Moreover, to facilitate ranking of security events so that security administrators can focus on the most security critical events first, we develop a prioritization scheme for incident classes and criticality ranking for network zones and asset types that reflect common process control system characteristics.

Incident Classes

Intrusion detection components can potentially report a very large number of alert types. Snort alone, for example, may be equipped with thousands of attack signatures. To handle this, we have provided a map of EMERALD reports and alerts from other components within DATES to a much smaller number of *Incident Classes*. Using incident classes facilitates the development of general correlation strategies and performing cross-sensor correlation. Specifically, the incident class abstraction enables one to specify correlation criteria at a higher level, resulting in a more extensible and reusable correlation system. Moreover, based on incident classes, SIEM systems may be able to combine reports from multiple detection components that use different event names to refer to the same attack, facilitating multi-sensor correlation.

Specifically, we employ the incident classification developed in our previous work on alert correlation [MCorr]. The list of incident classes is as follows (in alphabetical order):

- ACCESS VIOLATION: Attempt to reference, communicate with, or execute data, network traffic, OS services, devices, or executable content, in a manner deemed inconsistent with the sensor's surveillance policy.
- ACTION LOGGED: Diagnostic that is not necessarily indicative of malicious activity but is logged for potential future forensic use, such as a message that a device has responded to a particular Modbus request.
- ASSET DISTRESS: Indicative of a current or impending failure or significant degradation of a system asset (such as a host/port combination).
- BINARY SUBVERSION: Activity indicative of malicious code, Trojan Horses, or viruses. The most common attacks in this incident class are a variety of buffer overflows and code injections.
- CONNECTION VIOLATION: Attempt to establish a connection that is not allowed under the current policy.
- DENIAL OF SERVICE: Attempt to block or otherwise prevent legitimate access to a system resource, including host, application, network service, or device.
- EXFILTRATION: Attempt to export sensitive data through an unexpected or unauthorized communication channel.
- INTEGRITY VIOLATION: Attempt to destroy memory, data, or executable content in a manner inconsistent with the sensor's surveillance policy.
- PRIVILEGE VIOLATION: Theft or escalation of access rights to the level of system or administrative privileges.
- PROBE: Attempt to gain information on assets or services within the monitored domain. Refers to a variety of host and port scanning activity on the target network.
- SUSPICIOUS USAGE: Indicative of activity that is sufficiently unusual or suspicious, but not attributable to another incident class.
- SYSTEM ENV CORRUPTION: Unauthorized attempt to alter the operational configuration of the monitored system.
- USER ENV CORRUPTION: Attempt to alter the environment configuration of a user account.
- USER SUBVERSION: Attempt to gain the privileges of a locally administered account, potentially indicative of masquerading.

Prioritization of Incident Classes

To perform alert ranking, we developed a prioritization scheme for incident classes, based on the relative importance among the security objectives for process control systems. Generally, asset owners consider availability as the most important security objective, followed by integrity, and then confidentiality.

We group the incident classes into four "super classes" and assign severity values to them to reflect their importance for control systems. For example, the super class for "denial of service" and "asset distress" has the highest severity value, as the events in this class may affect the availability of target assets. At the other end of the

spectrum, the super class containing "action logged" and "probe" is typically less important. The "action logged" class tends to be informative, and pertains to events that are more suitable for postmortem analysis. Moreover, depending on the locations at which one observes events belonging to "probe" class, they may correspond to "background attack traffic", and usually do not affect the operations of the target assets. We note that the severity of the incident classes is only one of the factors used for alert ranking, and we will describe the other factors later in this report.

Table 1 summarizes the prioritization of incident classes: the higher the values, the more severe the events in the classes.

	Incident Class	**Numeric Severity**
Class 1	Denial of Service	4
	Asset Distress	
Class 2	System Env Corruption	3
	Integrity Violation	
	Binary Subversion	
	Privilege Violation	
Class 3	Suspicious Usage	2
	User Subversion	
	User Env Corruption	
Class 4	Access Violation	1
	Connection Violation	
	Probe	
	Exfiltration	
	Action Logged	

Table 1. Prioritization and classification of incident classes

Asset Types

The criticality of the targets is another factor that affects the importance of events. We use two attributes of assets to determine their criticality, namely, asset types and network zones in which the assets are located. Examples of asset types are historians and RTUs. We assign different weights to different asset types to reflect their criticality. Like incident classes, asset type presents a high-level abstraction that enables one to specify general correlation criteria for entire classes of assets as opposed to those for specific asset instances. As a result, we can use asset types to specify a more manageable and extensible set of correlation heuristics. Table 2 denotes the asset types and the associated criticality values used in the DATES project.

Asset Type	Criticality	Numeric Criticality
Remote Terminal Unit (RTU)	Very High	5
Front End Processor	High	4
ICCP Host	High	4
HMI Server	Medium	3
HMI Client	Low	2
Historian Server	Low	2
Historian Client	Very Low	1

Table 2. Criticality of asset types

Network Zones

Network zones present another dimension for specifying the criticality of assets. We use the heuristic that network zones closer to the field devices are more critical than those that are farther away. This is based on the following observations. Field devices interface with and may change the behavior of physical systems, and thus their availability and correct operations are most critical. Moreover, modern field devices may be able to function correctly (at least on a short-term basis) even when the control network assets such as FCP are unavailable. Process control networks with properly configured firewalls may restrict the possible attack paths from an external network to the field devices. For example, an adversary may need to compromise a machine in the DMZ and use it as a stepping stone to access the control network. Assuming the attack target is in the field network, network zones that are closer to the field network are more valuable from the adversary viewpoint. Table 3 denotes the network zones and the associated criticality values used in the DATES project.

Network Zone	Criticality	Numeric Criticality
Field	Very High	5
Control	High	4
DMZ	Medium	3
Corporate	Low	1

Table 3. Criticality of network zones

We have built a knowledge base into SIEM that comprehends criticality of cyber assets. As discussed above, field assets are assigned the highest priority, control network is considered high, DMZ medium, and corporate low. SIEM is also aware of incident class severity. Using this knowledge, we developed correlation rules that correlate events pertaining to attacks and perform alert ranking for correlated events.

Correlation Techniques

We present several correlation techniques employed in the DATES project for alert volume reduction, coordinated cross-site attack detection, distributed coordinated attack detection, and attack scenario detection based on zone criticality.

Common-field-value Aggregation

We combine security events that have the same values for a specified set of fields. A main application of this technique is to reduce the number of alerts from an intrusion detection sensor for an attack. An example is that we may aggregate alerts generated by a sensor if they have the same values for source IP address, target IP address, and event name within a specified time window. When an attack (e.g., port scanning attack) triggers thousands of alerts for each target host, this technique may reduce them to one correlated alert.

Increased Rate Detection

To detect coordinated attacks against multiple sites (such as attacks targeting a specific sector), we developed a technique called *increased rate detection*. The intuition is that individual sites may observe various security events from time to time. Thus, using a naive approach that correlates the presence of events pertaining to a specified incident class among various sites may lead to spurious correlation results. To address this issue, we consider the rates of events to reduce the probability of false alarms for cross-site correlation. Our technique will declare coordinated attacks when multiple sites observe an increased rate of events that belong to the same incident class.

Alert Prioritization for Control Systems

Tailored to the environment of electric utilities networks, we also raise alarms when events involve certain critical parts of the networks. These include raising alarms when events originate from inside the utility's networks or when they pertain to critical asset types or zones.

Distributed Coordinated Attack Detection

In a distributed coordinated attack, multiple sources are involved in attacking a target host or service in a coordinated manner, typically within the same time window. An example of distributed coordinated attack is the distributed denial-of-service (DDoS) attack, whose goal is to cause the target host or service to be unavailable. In DATES, we implemented an alert correlation rule set in ArcSight to detect DDoS attacks.

Event Chaining for Zone-based Criticality Escalation

This technique attempts to chain together events belonging to multi-step attack scenarios that progress from a low-criticality network zone to one of higher criticality. The key idea is to take advantage of the observation that process control networks tend to have predictable and "layered" communication patterns among the hosts in different zones. With properly configured firewalls, an external adversary with potential access only to the corporate network may need to

compromise machines through a series of network zones (from the DMZ to the control network) before gaining access to high-value targets in the field networks.

Our approach enables correlation and visualization of an attack as it crosses zones in the process control network. Recalling the criticality values for zones discussed earlier in the report, we assign the highest priority to assets in the field zone, high priority to the control zone, medium priority to the DMZ, and lowest priority to the corporate network. As shown in the algorithm of Figure 2, IDS alerts pertaining to zone- or utility-crossing events may be correlated based on the criticality of the zones pertaining to the source and destination IP addresses and matching of the source IP address in an alert with the destination IP address of another alert.

Given a set of alerts A0, A1, A2, and so on, the zone-based criticality escalation algorithm will correlate them as an event chain if the following conditions are met:

```
  (1) zone(dst(Ai)) is “internal”
  (2) dst(Ai) = src(A(i+1))
  (3) criticality(zone(src(Ai))) ≤ criticality(zone(dst(Ai))) OR
      zone(src(Ai)) is “external” OR
      utility(zone(src(Ai))) ≠ utility(zone(dst(Ai)))
where
   dst(A) returns the destination IP address of alert A,
   src(A) returns the source IP address of alert A,
   zone(X) returns the zone to which IP address X belongs,
   zone Z is “internal” if it is monitored by a participating IDS,
   zone Z is “external” if it is not internal,
   criticality(Z) returns the criticality of zone Z, and
   utility(Z) returns the identifier of the utility to which zone Z
   belongs.
```

Figure 2. Code for escalation of criticality

Condition (1) establishes the boundary case that the event chaining process stops when the destination zone of the last event is no longer being monitored. Condition (2) corresponds to the requirement about matching the destination IP address of an event with the source IP address of another event. Condition (3) pertains to the criterion about non-decreasing zone criticality (i.e., the zone corresponding to the destination IP address of an event should be at least as critical as the zone corresponding to the source IP address). There are two exceptions for this criterion: one for the case that the source is external, in which case the numerical value of the criticality is undefined, and one for events between two different utilities.

ArcSight Implementation

We implemented the detection and correlation techniques introduced above by using the ArcSight SIEM system. Our intrusion detection systems feed events into ArcSight using the Common Event Format (CEF) and ArcSight's Smart Connector technology. Within CEF, we are using the "Device Event Category" field to contain the incident class of an event.

Common-field-value Aggregation

Each connector allows a configuration for field-based aggregation. Figure 3 shows the values we use to perform such aggregation.

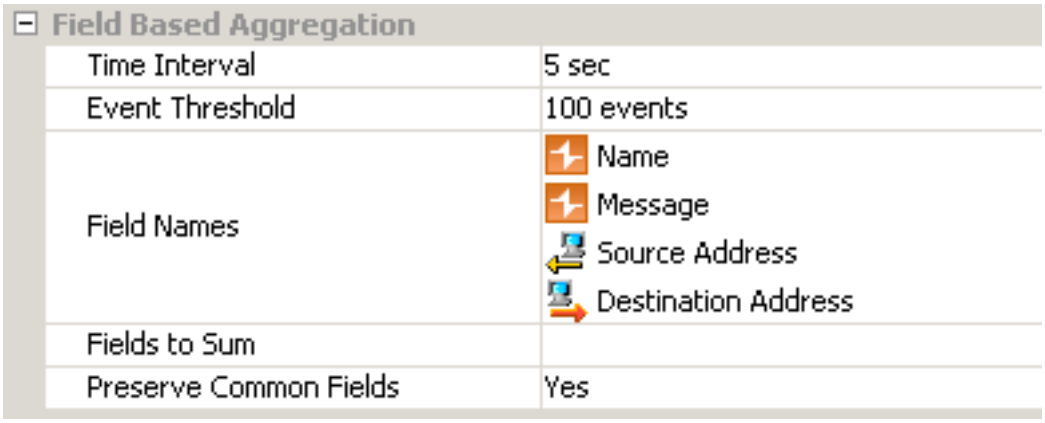


Figure 3. ArcSight definition of field-based aggregation

This configuration aggregates as many as 100 events within 5 seconds into one event if the four fields Name, Message, Source, and Destination Address are the same. All other fields that are the identical in the aggregated events are preserved in the newly created event.

Alert Prioritization for Control Systems

We have implemented a number of rules to elevate certain events that are of importance in a control system.

We raise alarms when we encounter events with incident classes from group 1 or 2 that have a destination address of an asset with criticality ≥ 4 or in a zone with criticality ≥ 4. We assign the newly created alarm event a priority of 10, if the asset type or the zone has criticality 5; otherwise, the priority is 9.

Furthermore, events that pertain to an internal source and destination are further subject to a correlation as a potential event chain as discussed below.

Ignoring the destination of events, those that originate from a more critical part of the network are also of concern. We fire alarms when encountering events with the source address located in the control or field network zones. We assign the newly created alarm a priority of 10 if the event emanates from the field network as the more critical part of the network, and a priority of 9 if it came from the control network.

Finally, we are also specifically looking for events pertaining to malformed ICCP traffic. Here, we are looking only for events with names matching the snort identifiers "1:1111401" through "1:1111409."

Increased Rate Detection

We use ArcSight's built-in statistical data monitors to calculate rate changes. One example of measuring the moving average is shown in Figure 4. We restrict this data monitor to look only at events with incident classes that came from the connector located at the testbed at Sandia, which represents one utility network. It takes the average of the aggregated event count after performing field-based aggregation at each connector, so one event may refer to multiple aggregated events. Measurements are taken separately for each incident class, which is stored in the "Device Event Category" field. The sampling interval is 120 seconds, and measurements require five samples to calculate an average value. The variables "s" and "c" store the average value over the past samples and the current value, respectively.

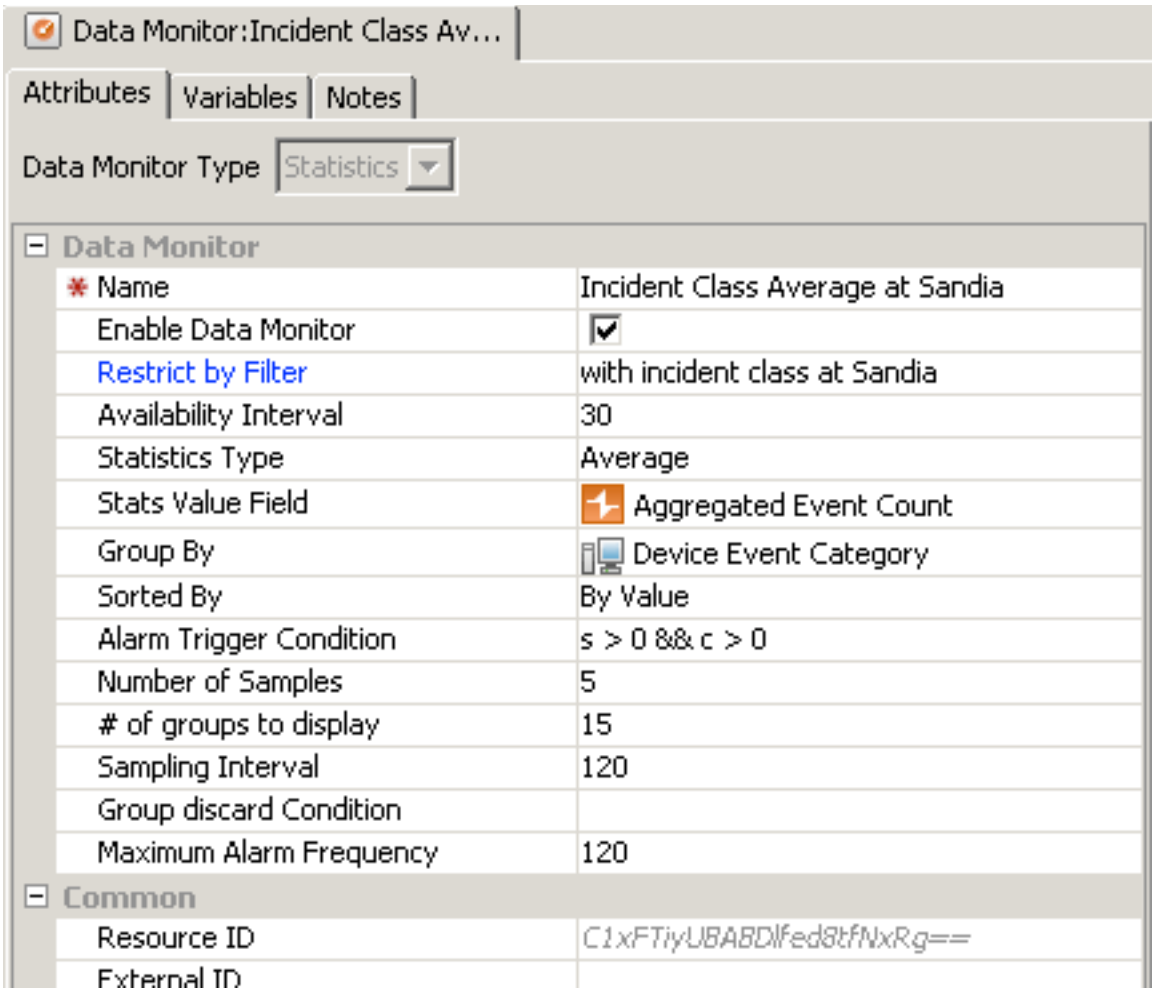


Figure 4. ArcSight definition of increasing incident class average

To detect an increased average of any incident class, we created a rule to match the events that the statistics data monitor emits. Figure 5 illustrates the conditions of this rule, and Figure 6 defines the variable "ratio".

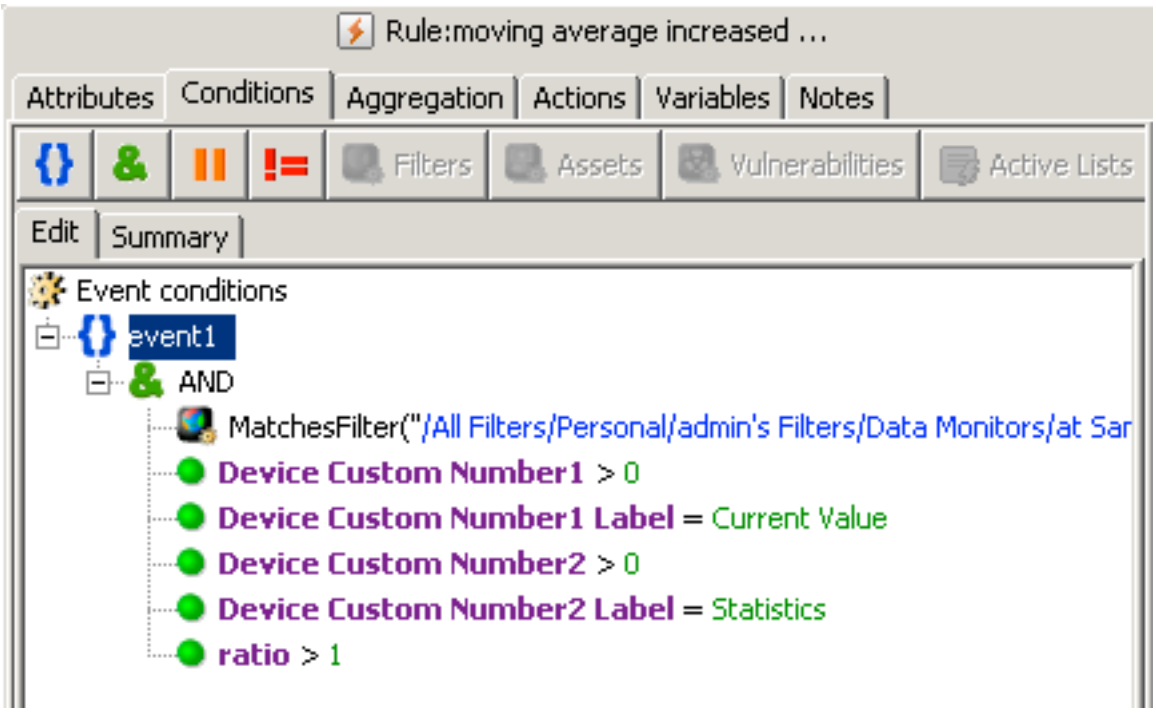


Figure 5. ArcSight rule to match meta-events from the moving average data monitor

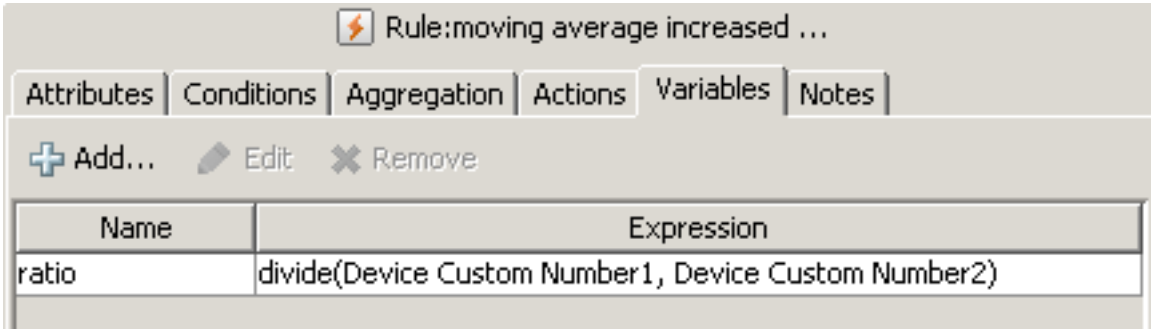


Figure 6. Definition of ratio for the moving average rule

We essentially match all events that come from a specific data monitor using the ArcSight internal "Resource ID" and check that it defines custom number fields 1 and 2 (Figure 6), which should contain the "Current Value" and "Statistics" (the "c" and "s" variables in Figure 4). Finally, we fire this rule only if the ratio defined as the division of c by s is greater than one, which means that the current value is larger than the past average. As a result of firing this rule, we add the incident class to an active list called "Increased Incident Classes", which is used in turn to implement the state diagram shown in Figure 7.

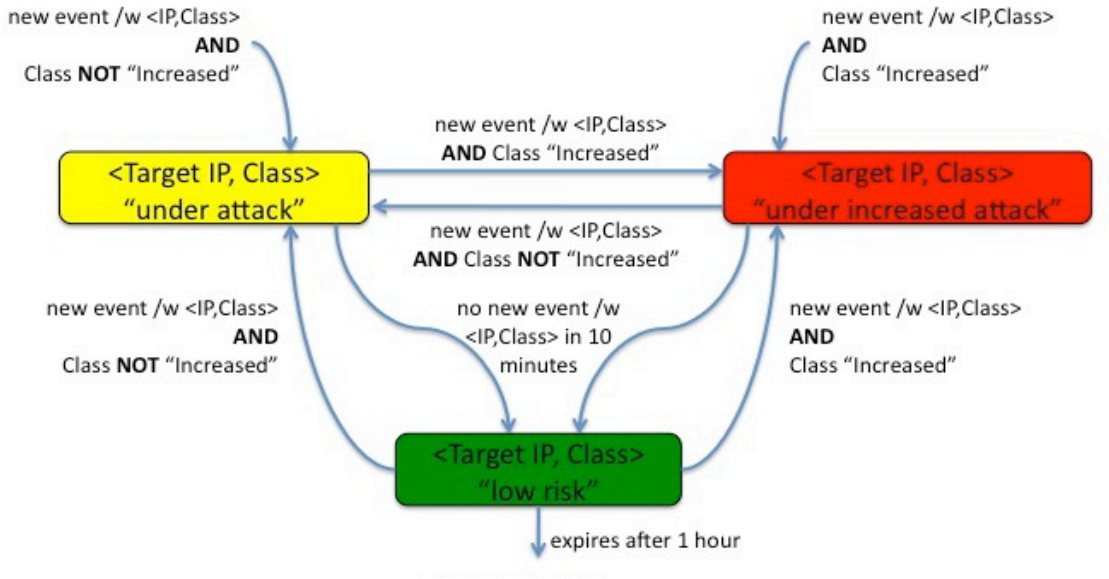


Figure 7. State transition diagram relating target IP attack state and incident class

Separately from monitoring the average rate for each incident class, we are also collecting all target IP addresses from any event that contains a valid incident class. If such an event arrives and the corresponding incident class is not in the list of currently "Increased Incident Classes", we label the pair of IP address and class as "under attack" and associate a medium threat level (yellow) with it. If the incident class is in the list of "Increased Incident Classes", we label the pair "under increased attack" and assign a high threat level (red) to it. This classification expires after 10 minutes and the pair is then labeled "low risk" with a low threat level (green) unless another event with the same target IP address and incident class arrives.

We have also added a rule to detect when an incident class is detected as "increased" at both monitored utility sites. This rule fires when a new entry is added to the active list with increased rate incident classes at one of the two sites while the same entry is currently present at the respective other site. As an action from this rule, a dialog box (Figure 8) opens to alert the human operator. Other means of notifying appropriate personnel, such as email and pager messages, can be used as well.

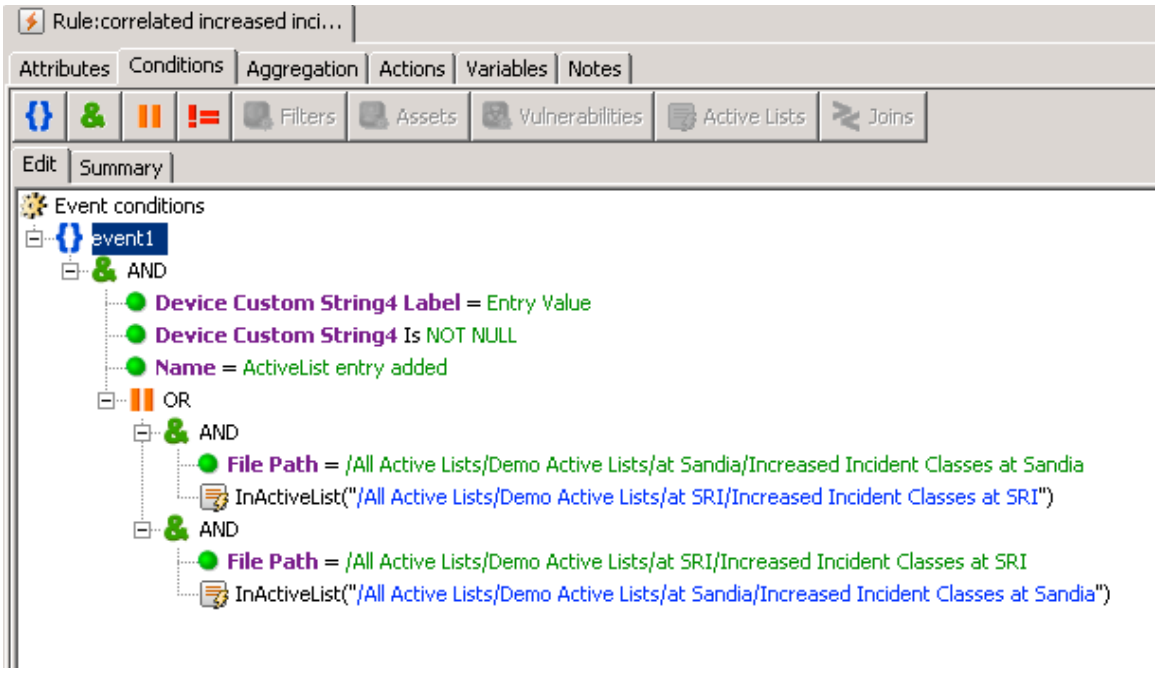


Figure 8. Alert Window for Increased Incident Event

Distributed Denial-of-Service Attacks

In this section, we report on our implementation of detecting distributed denial-of-service (DDoS) attacks. First, we require multiple events with the same destination but different sources. We also require these events to be of type "Base" or "Aggregated" (to ignore meta-events generated from rules) with a destination inside the utility networks. In addition, we test that the destination is not already flagged as a potential DDoS target (i.e., is in the Active List called "current DDoS targets"). We also prevent matching when a prior DDoS correlation with the same target has occurred during the time window using a negated event feature.

This initial rule to match two events is split to cover three different cases. These cases are designed to be mutually excluding to prevent ArcSight from matching more than once, as we are also counting the number of events for display in more elaborate dashboards.

1. One more severe (which could be Asset Distress) and one less severe event
2. One Asset Distress and one more severe event that is not Asset Distress
3. Two more severe events that are both not of class Asset Distress

By defining events by these three cases, we can address double counting when the matched events are symmetric. ArcSight would normally fire the rule twice; therefore, we also impose ordering of event IDs in this case. Another reason for establishing these cases is that the position of an Asset Distress event should always be "event2".

Figure 9 shows the conditions of the rule to match an initial DDoS attack of the first case above.

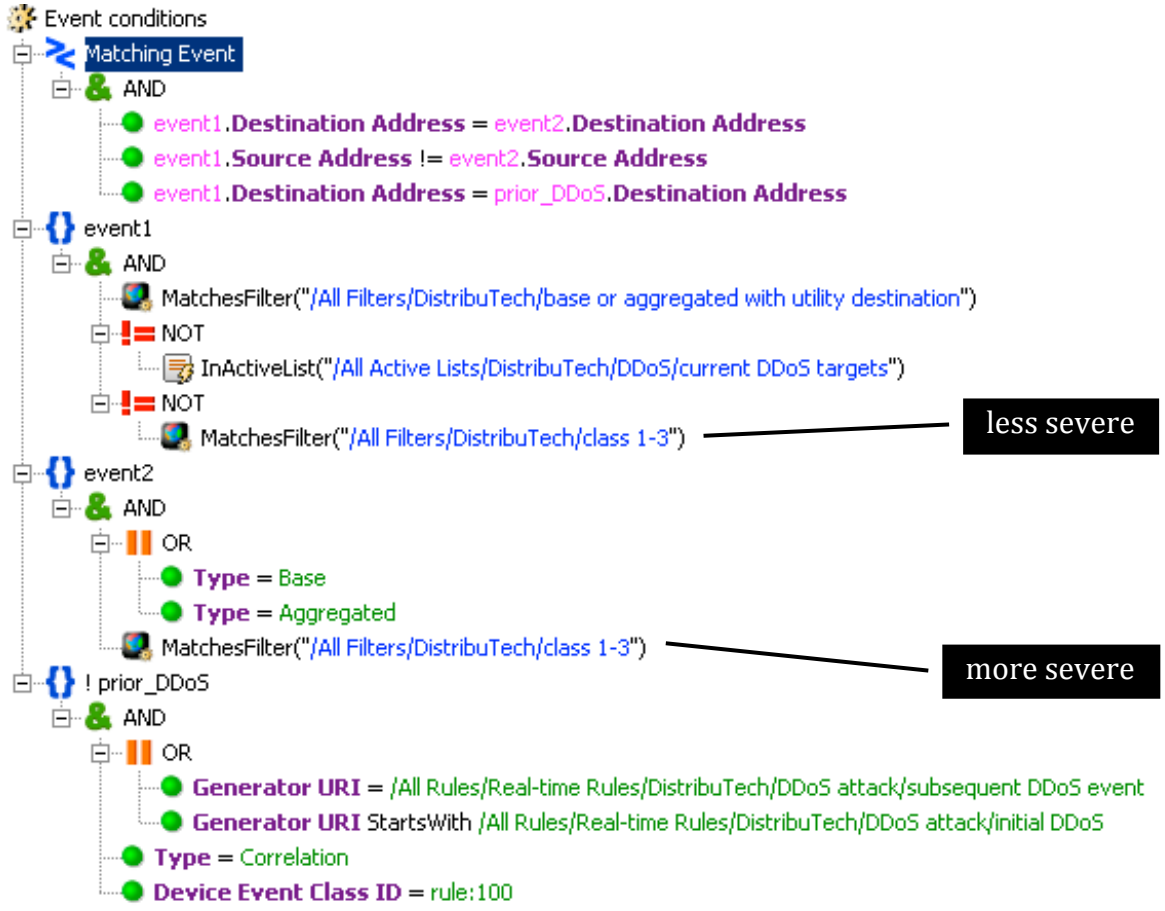


Figure 9. ArcSight Rule definition for DDoS, Case 1

Figure 10 shows the second case, which is almost identical except for the matching of the incident classes in “event1” and “event2”.

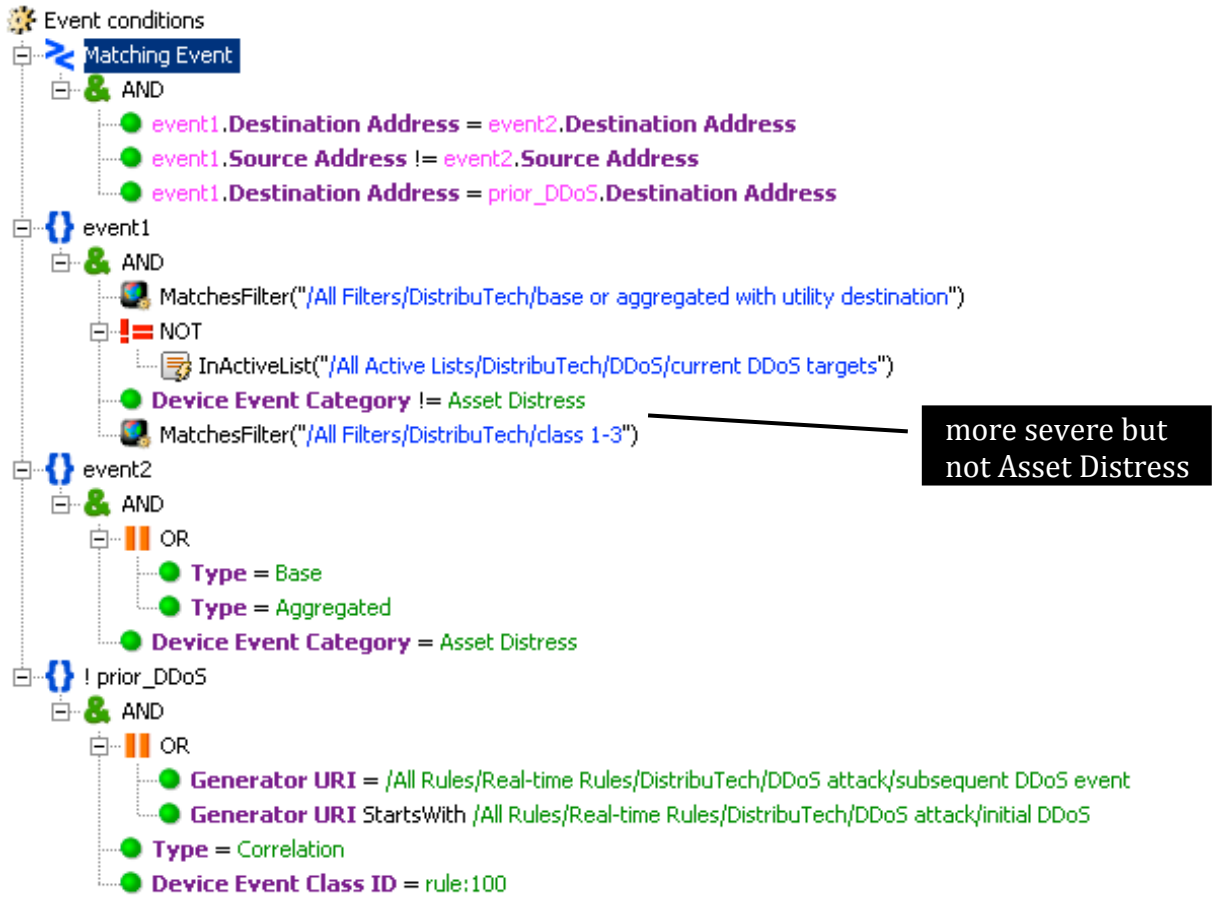


Figure 10. ArcSight Rule definition for DDoS, Case 2

Figure 11 shows the third case, in which we impose an ordering of the event IDs to prevent the otherwise symmetrical events from matching each other. Note that the filter called "base or aggregated with utility destination" is in effect the same as the condition below in "event2" that requires the type "base" or "aggregated" along with the join condition of having the same destination address as "event1" (which must be inside the utility networks).

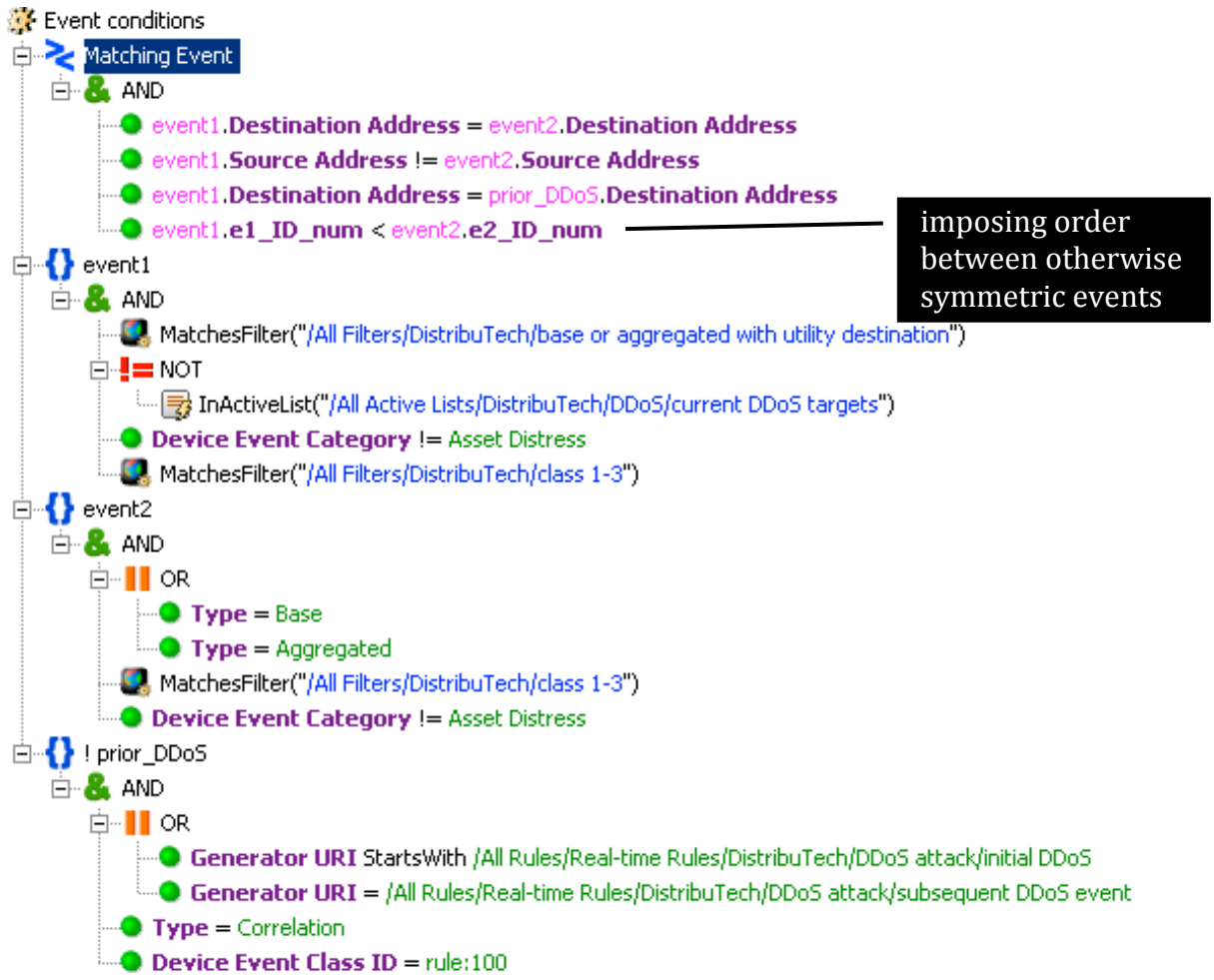


Figure 11. ArcSight Rule definition for DDoS Case 3

Each of theses rules that matches an initial (potential) DDoS attack add the destination IP address to an Active List called "current DDoS targets. Subsequent events that have a destination IP contained in this list are then correlated using event graphs.

Attacks that Traverse Network Zones

To detect attacks that unfold as a series of events, which progress through the networks, we first define a filter to capture potential candidates. Figure 12 shows the definition of such a filter in the ArcSight SIEM, which looks for a base or aggregated event with a destination inside the utility networks and equal or increasing criticality between source and destination. We define an event as progressing in criticality if either a) the numeric criticality of the source zone is less than or equal to that of the destination zone, or b) the source is outside of the utility networks, or c) the event crosses from one utility to the another.

Figure 12. ArcSight filter for potential network traversal events

The next challenge is to formulate rules that fire when potential candidates of events form a chain in which the source of a new attack step is the target of a previous attack step. To implement this chaining of events, we decided against a simple recursive solution as it poses the danger of running in loops. Instead, we spelled out a rule for each *i*th chain element starting with an initial rule "chain2" to match the first two events and then matching existing chain events "chain*n*" with one event to extend the chain by one. In our prototype implementation we count up to "chain6" to match chains of length six in our prototype implementation. In an actual deployment, we suggest creating rules to a higher number. One reason for this approach is the power of the rule-matching engine in ArcSight. A rule that is triggered creates a meta-event that enters the incoming event stream and could be consumed by any rule as if it were an event created from one of the sensors connected to ArcSight. We use this feature cautiously in our implementation, as a poorly written rule may easily create a loop or consume too much memory and processing power when under heavy load, which in turn may cause ArcSight to automatically disable such rules in order to maintain operation of the whole system.

Figure 13 shows the definition of rule "chain2" in ArcSight. Recall that the goal of this rule is to fire when two events are initially identified as a chain of length two. The conditions define two events e_1 and e_2 that are a chain but not a loop, i.e., $dst(e_1) = src(e_2)$ and $src(e_1) = dst(e_2)$. Both events must match the filter "progressing base or aggregated with utility destination" defined in Figure 12.

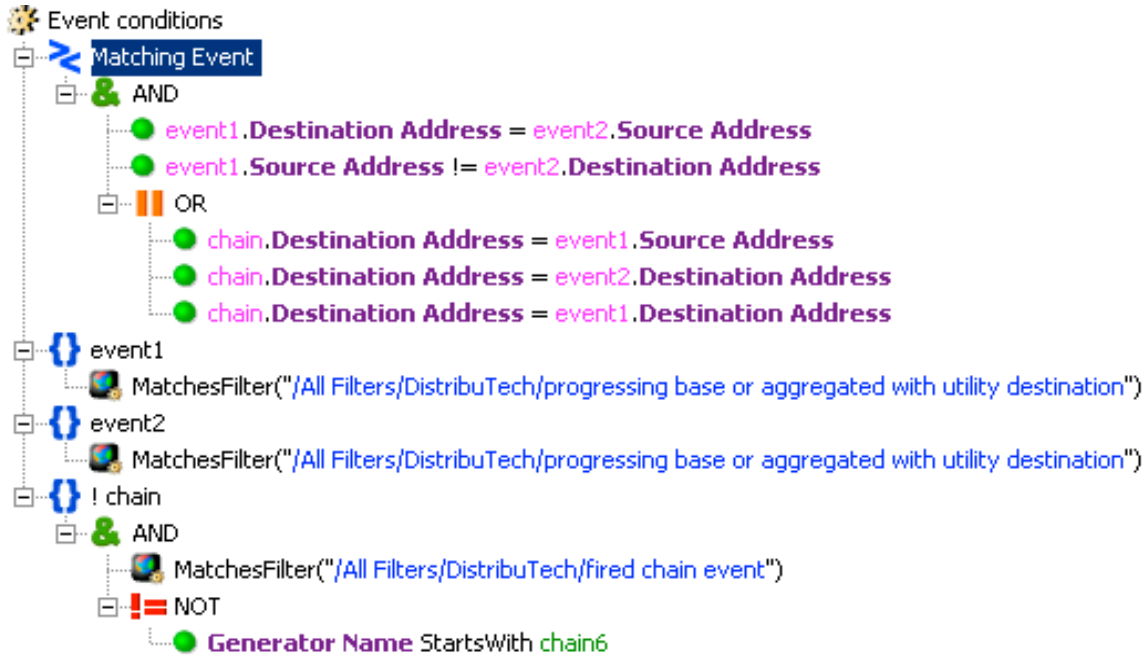


Figure 13. ArcSight event chaining rule "chain2"

In addition to this positive formulation of what it means to declare an initial chain of length two, we employ the concept of negated events in ArcSight to prevent the rule from firing in situations when a chain already exists to which one of the events e_1 or e_2 could be attached. Using a negated event requires the absence of any matching event at the time the rule is evaluated. Here, the negated event c denotes a prior firing of any of the "chain2" through "chain5" rules ("chain6" is explicitly exempt, as it is the last chain rule in our prototype implementation). Then, in the JOIN condition (labeled "Matching Event") at the top of the tree, we prevent the rule from firing in the following three situations as depicted in Figure 14. If there was a previously fired chain event c_n of length n, and its destination matches one of the three locations of the currently matching "chain2" events, namely $src(e_1)$, $dst(e_1)$, and $dst(e_2)$, then we want to prevent the establishment of a new chain of length two. Instead, a different rule for chains of length n+1 could possibly match with the respective event e_1 or e_2 to extend the prior chain event c_n. Thus, we exclude the longest possible chain rule in the negated event ("chain6" in our prototype implementation) as no chain rule for length n+1 exists. In this case, it is prudent to have "chain2" fire and start a new chain of length two.

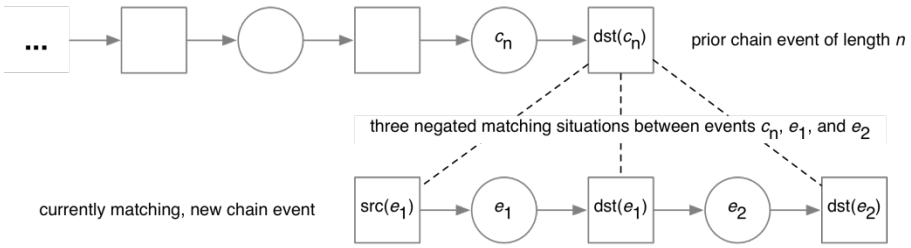


Figure 14. Three negated matching situations for "chain2"

We then define the rules “chain3” through “chain6” as follows; we will use “chain3” as shown in Figure 15 as an example. Let us assume that we attempt to match a chain of length n. In the example of “chain3” with n=3, we match a prior chain event c_2 that refers to a chain of length n-1=2 and means that the corresponding rule has fired and created a meta-event in the event stream with another “progressing base or aggregated with utility destination” event e_3. Again, these two events must form a chain but not a loop, i.e., $dst(c_2) = src(e_3)$ and $src(c_2) = dst(e_3)$. Then, analogous to the rule “chain2” we also require the absence of another chain event with length $\geq n$ using the concept of a negated event c. The logic of applying the negated event is analogous to the logic explained above in relation to “chain2” rules and depicted in Figure 14.

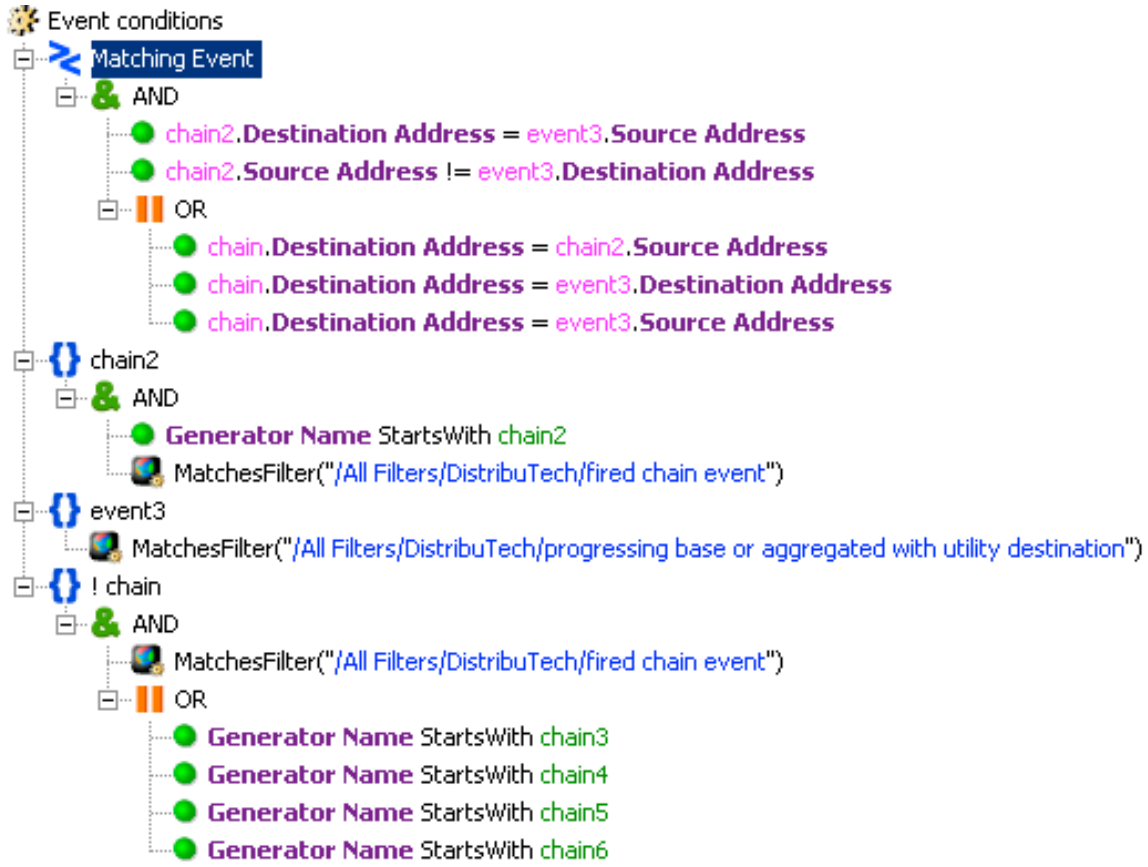


Figure 15. ArcSight event chaining rule "chain3"

Cross-Site Attack Correlation

Our approach to cross-site correlation applies to two situations. The first is to correlate attacks consisting of similar events against similar targets or zones at two utilities occurring closely in time. The notion of "similar" is based on incident class. We also look for common origin. The correlated attack report lists assets in a generic way, such as "Control Zone HMI". By abstracting details from IP addresses and specific architectures to zones and device types, we do not reveal sensitive details of operational networks and security posture.

The second cross-site correlation considers an apparent attack from one utility to another over some protocol such as ICCP. Note that the zone traversal attack included an attack from the control center of one utility (specifically, the ICCP client in that zone) to the second utility. In this case, presumably the attacker gains control of the ICCP client or server at the source utility, and there may be events related to that attack, which the security officer at the utility should then be instructed to investigate.

DATES Testbeds

The DATES project implemented two testbeds for deployment, test, and experimentation. These testbeds were located at SRI and Sandia National Laboratories. For the period of experimentation beginning in December 2009, the two testbeds were linked using a secure tunnel connection, so that alerts from the Sandia testbed could be processed by the ArcSight installation at SRI. This permits simultaneous visualization of attacks at both sites, and was the distributed test platform on which we built our multi-site detection and correlation experiments.

Invensys Distributed Control System (DCS) Testbed

The SRI test environment is based on a DCS from Invensys Process Systems, IA series [Invensys]. This system has the following key elements:

- Application workstation (AW) for configuration, visualization, and control. This is dual homed with a connection to a control LAN as well as an external interface.
- Control LAN based on a redundant pair of Enterasys switches (optical Ethernet).
- Invensys Field Control Processor (FCP) module.
- Field bus that connects the FCP to (presently) two Ethernet Field Bus Modules (FBM).
- Field LAN connecting the FBM, simulated Modbus devices (Modbus simulators from Modbustools.com and Calta) running in virtual machines.
- Monitoring monitoring system that connects to the control LAN (AW, switches, and FCP) and separately to the field LAN (FBM and devices).
- Interfaces between the monitoring system and the ArcSight Security Information and Event Management (SIEM) platform.

The test environment system is shown in Figure 16. The protocol on the control LAN, between the AW and the FCP, is proprietary. The protocol on the field LAN, between the FBM and field devices, can be any of a number of common industrial protocols, and we chose Modbus in our experiments.

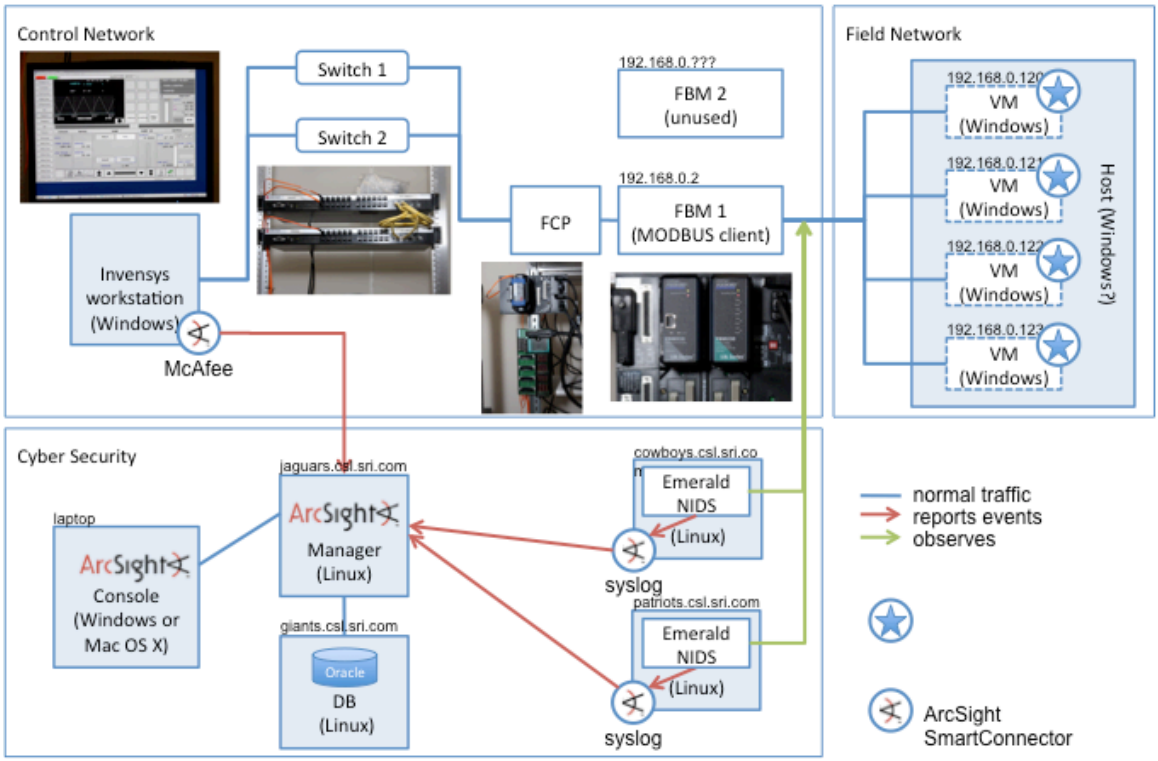


Figure 16. Invensys demonstration DCS at SRI International

Virtual Control System Environment (VCSE)

In addition to the Invensys DCS at SRI, Sandia implemented a DATES testbed based on the Sandia Virtual Control System Environment (VCSE)[SNL 2010]. The VCSE complements the SRI system and permits a richer emulation, including corporate zone and DMZ emulation, as well as greater flexibility in configuration of assets.

VCSE models represent the relevant portions of cyber-physical systems and their threats. They are instrumented to facilitate the analysis of the physical effects that the threats may have on the systems under study. The models are constructed from real, emulated, and simulated components that are vulnerable to actual, representative, and simulated malware and other hostile actions. Emulators duplicate (provide an emulation of) the functions of one system using a different system, so that the second system behaves like (and appears to be) the first system. For purposes of this discussion, emulators include software-based emulators that emulate real computer hardware as well as simulation models that are configured to emulate computers and physical equipment. VCSE simulations are generally built as computer models. Most of the simulations have emulation interfaces to behave, as seen from the outside, as an emulator.

Sandia's VCSE is, by its nature, a distributed tool-oriented environment. VCSE instantiations or models vary by the specific tools brought together and their configurations.

Generally, Sandia's VCSE models combine real or representative SCADA systems, a mixture of real and emulated network components, emulated control interfaces, and simulated physical plant models. Cyber threats are represented with actual or representative malware. Physical threats are represented in the physical model, and the analysis team members typically play the part of insiders and cyber terrorists.

Sandia typically uses virtual machine (VM) technologies to host real system components. In this way, for example, analysts can run a small network of SCADA tools on one host computer.

Researchers at Sandia and SRI International instrumented a VCSE to develop and test control system intrusion detection, SIEM, and large-scale threat analysis technologies. Sandia provided SRI with a variety of VCSE models with threats that would be difficult to detect using existing IT security technologies. The instrumented VCSE was exercised in various modes of normal operation as well as a variety of attack scenarios. The team established a secure communication channel to the SRI facility to permit a unified view of threats against two simulated utilities.

Figure 17 shows the configuration of the VCSE-based testbed hosted at Sandia.

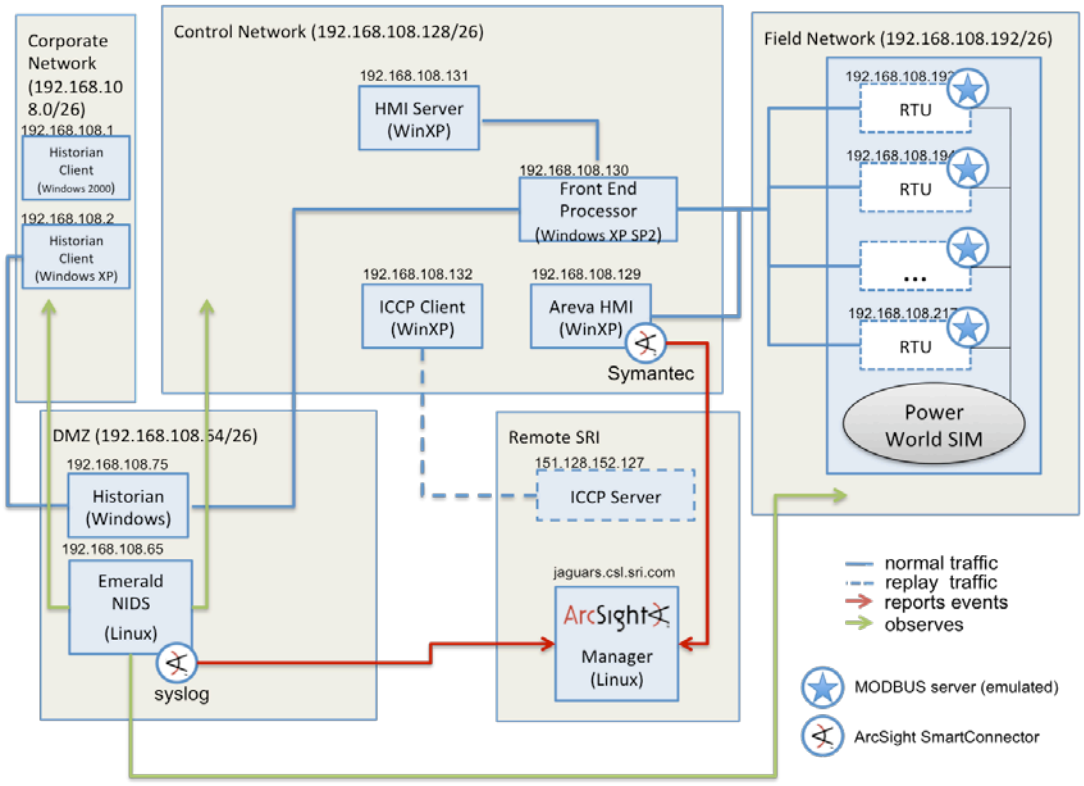


Figure 17. DATES testbed using Sandia's Virtual Control System Environment

If the tool were a real cyber-physical system, the analysts would need to significantly restrict the level of cyber attack that they could launch on the system. This is not the case with VCSE systems, where even with the effects of attacks that

destroy equipment, computer operating systems can be regenerated in a matter of seconds and the system restarted instantly for further studies. To determine how the security systems respond to zero-day attacks and other anomalies, Sandia modelers can introduce new attacks for these tests.

Multi-site Testbed Setup

The multi-site testbed setup uses the SRI and Sandia test facilities, connected over a secure connection, simulating two different utilities. Some services typical of inter-utility connections, such as client-server traffic over the Inter Control Center Protocol (ICCP), are simulated by packet replays. The multi-site experiment seeks to demonstrate attack detection and correlation at each utility, as well as correlation of DDoS and an attack from one utility to another over ICCP. The multi-site experiment setup is shown in Figure 18.

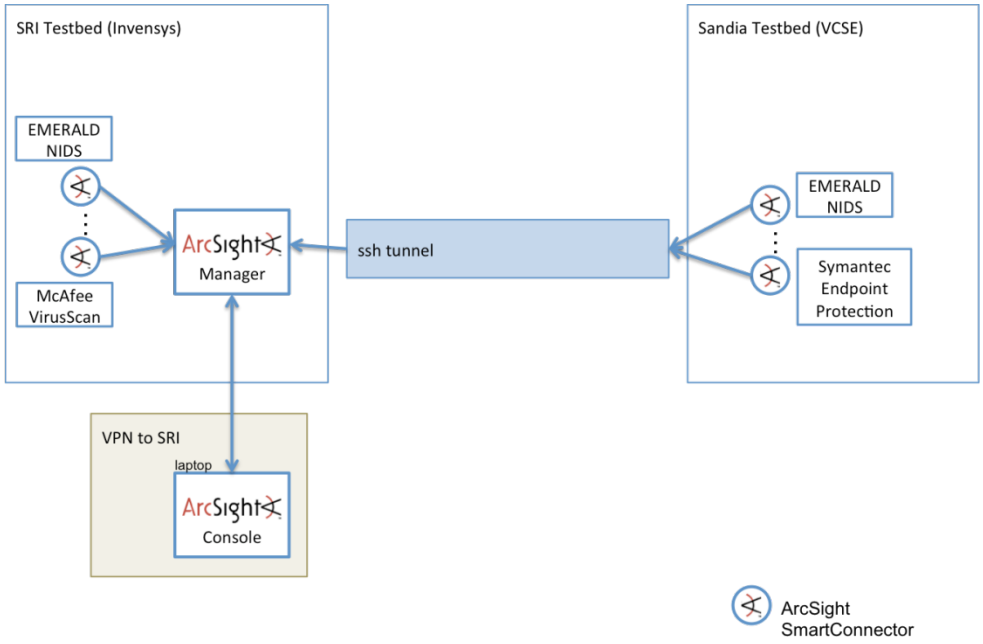


Figure 18. Multi-site Configuration

Validation

Sandia devised two different attack scenarios for the testing and validation of DATES and to be used for demonstrations. The first scenario is a DDoS in which a critical asset in a utility is targeted by a number of other compromised systems. The second scenario is series of attack steps that demonstrates an attacker that originates in a lower-priority network zone and is able to traverse across zones affecting higher-priority critical assets. The attacker is even able to jump into a second interconnected utility. Both scenarios are intended to validate a wide range of DATES detection capabilities.

Distributed Denial of Service Scenario

A DoS of a critical asset in a utility poses a great risk. A DDoS is often used as an effective means of creating a DoS on a particular asset. This is accomplished by having several systems participating in a coordinated attack in order to overwhelm the target. The participating systems in the attack are, most likely, themselves victims from a prior compromise. In this scenario, it is assumed that the systems participating in the attack have been previously compromised and are hosting a malicious application that is programmed to run at a specific date and time. Specifically, the assumed compromised systems on the Sandia testbed (also referred to as Utility 1) are

- Two different historian clients in the corporate network
- A front end processor in the control network
- A human machine interface server in the control network

Access control lists in the firewalls allow all these systems to communicate with a historian server in the DMZ network. The malicious application located on the systems listed above will leverage the trusted access controls in order to execute a coordinated attack against the historian server. The malicious applications perform the following attacks:

- TCP SYN flood that leaves partially opened TCP connections on all network ports in order to overwhelm the network interface
- Flood of 'garbage' traffic to port 80 (the historian server provides a web interface on port 80)
- Flood of 'garbage' traffic to port 1000 (the historian receives database updates on port 1000)

The DDoS scenario concludes with the coordinated attack successfully causing an asset distress at the historian server. The historian server is no longer able to properly serve content over port 80 or receive updates over port 1000 because of the overwhelming amount of partially opened network connections and data flooding on those particular ports.

A similar setup is used to run the DDoS scenario at the SRI testbed (also referred to as Utility 2). In this case, an ICCP server is targeted by several other compromised systems.

Network and Utility Traversal Scenario

The second scenario demonstrates an active attacker on the Utility 1 network who is able to traverse starting at the corporate network, through to the field network, and then across to Utility 2. It is assumed that the attacker has access to the corporate network either as a malicious insider, or by coming in through an outside Internet

connection. Table 4 lists the attacker's actions at each step and the results of that action.

Step	Action	Results
1	Scan corporate network for interesting hosts	Attacker finds several hosts and identifies a historian client
2	Port scan historian client	Attacker discovers potentially vulnerable MS Windows services running on the host
3	Run several different Windows exploits against the interesting open network ports	A buffer overflow exploit worked on a Windows network service, and grants the attacker a remote session with the historian client
4	Monitor network traffic on historian client	Attacker observes network communication with a host (historian server) in the DMZ
5	Firewalls do not allow the attacker to access the DMZ from his system, but the compromised historian client can. Set up routes on the historian client so traffic is forwarded from the DMZ to the attacker's system	Attacker can now send and receive traffic to the discovered historian server in the DMZ
6	Assuming the historian is running the same unpatched version of MS Windows, attempt to rerun the same Windows exploit	The buffer overflow exploit worked again and grants the attacker another remote session with the historian server in the DMZ
7	Monitor network traffic on historian server	Attacker observes network communication with a host (FEP) in the control network
8	Set up routes on the compromised historian server so traffic is forwarded from the control network to the attacker's system	Attacker can now send and receive traffic to the discovered FEP host in the control network

9	Repeat Windows exploit on FEP	The buffer overflow exploit worked again and grants the attacker another remote session with the FEP in the control network
10	Monitor network traffic on the FEP	Attacker observes network communication with a host (RTU) in the field network
11	Set up routes on the compromised FEP so traffic is forwarded from the field network to the attacker's system	Attacker can now send and receive traffic to the discovered RTUs
12	Assuming limited knowledge of ModBus/TCP, the attacker attempts to send a ModBus command to the RTU, but with an invalid data value	Attacker is unsuccessful at getting the RTU to execute the command and moves on to other interesting targets
13	Further monitor network traffic on the FEP	Attacker observes network communication with a host (HMI)
14	Set up routes on the FEP to allow communication between the attacker and HMI	Attacker can now send and receive traffic to the discovered HMI
15	Repeat Windows exploit on HMI	The HMI is running Symantec End Point protection that identifies the attack and blocks it.
16	Further monitor network traffic on the FEP	Attacker observes network communication with a host (ICCP host)
17	Set up routes on the FEP to allow traffic from the attacker to the ICCP host	Attacker can now send and receive traffic to the discovered ICCP host
18	Repeat Windows exploit on ICCP host	The buffer overflow exploit worked again and grants the attacker another remote session with the ICCP host
19	Monitor network traffic on the ICCP host	Attacker observes network communication with an external host (ICCP server) located at Utility 2

20	Set up routes on the compromised ICCP host to allow communication from the attacker to the ICCP server at Utility 2	Attacker can now send and receive traffic to the discovered ICCP server at Utility 2
21	Attacker sends malformed ICCP messages to the ICCP server	A known vulnerability in some ICCP implementations will cause the ICCP server to crash on malformed ICCP messages. The attacker is successful at crashing the ICCP server, and Utility 1 and Utility 2 can no longer share power data

Table 4. Multi-step attack scenario

As Table 4 indicates, the attacker in this scenario is able to accomplish many objectives by performing several different types of attacks. The goal of this scenario was to validate several of the DATES detection capabilities including events from eBayes, eModbus, eFlowmon, Snort equipped with rule sets for monitoring enterprise networks and process control networks, and host-based intrusion detection systems.

Table 5 lists the events collected at the ArcSight SIEM with each step of the attack.

Step	**Action**	**Events Generated**
1	Scan corporate network for interesting hosts	None
2	Port scan historian client	Snort (122:1 – TCP Portscan)
3	Run several different Windows exploits against the interesting open network ports	eFlowmon – LEARN_NEW_FLOW
4	Monitor network traffic on historian client	NA
5	Firewalls do not allow the attacker to access the DMZ from his system, but the compromised historian client can. Set up routes on the historian client so traffic is forwarded from the DMZ to the	NA

	attacker's system	
6	Assuming the historian is running the same unpatched version of MS Windows, attempt to rerun the same Windows exploit	Snort (1:2002903 – ET Exploit) eFlowmon – LEARN_NEW_FLOW eBayes – NEW_SVC eBayes – UNUSUAL_TRAFFIC
7	Monitor network traffic on historian server	NA
8	Set up routes on the compromised historian server so traffic is forwarded from the control network between the attacker's system	NA
9	Repeat Windows exploit on the FEP	Snort (1:2002903 – ET Exploit) eFlowmon – LEARN_NEW_FLOW eBayes – NEW_SVC eBayes – UNUSUAL_TRAFFIC
10	Monitor network traffic on the FEP	NA
11	Set up routes on the compromised FEP so traffic is forwarded between the field network and the attacker's system	NA
12	Assuming limited knowledge of Modbus/TCP, the attacker attempts to send a Modbus command to the RTU, but with an invalid data value	Snort (1:3005205 – Illegal Data Value) Snort (1:3005609 – Modbus Illegal Value) eModbus – NEW_MODBUS_FUNCTION…
13	Further monitor network traffic on the FEP	NA

14	Set up routes on the FEP to allow communication between the attacker and HMI	NA
15	Repeat Windows exploit on HMI	eFlowmon – LEARN_NEW_FLOW Symantec – MSRPC Service BO
16	Further monitor network traffic on the FEP	NA
17	Set up routes on the FEP to allow traffic between the attacker and the ICCP host	NA
18	Repeat Windows exploit on ICCP host	eFlowmon – LEARN_NEW_FLOW
19	Monitor network traffic on the ICCP host	NA
20	Set up routes on the compromised ICCP host to allow communication between the attacker and ICCP server at Utility 2	NA
21	Attacker sends malformed ICCP messages to the ICCP server	Snort (1:1111407 – COTP Protocol Error) eBayes – SVC_DOWN eFlowmon – LEARM_MISSING_FLOW

Table 5. ArcSight events corresponding to attack scenario

ArcSight Output

Implemented scenarios in the ArcSight security event information management system can be visualized, as shown in Figure 19.

The figure provides a screen shot of the statistics data monitors that observe the rates of all incident classes in the incoming event stream from each testbed (SRI on the left and Sandia on the right). The yellow lines refer to the moving average, while the green spikes denote the current measurement for that time window. We see large numbers of "Probe" events and smaller numbers of "Action Logged" events at

both sites. At SRI, during this experiment we have also detected "Connection Violation" events, whereas at Sandia, events of type "Suspicious Usage" and "Binary Subversion" occur.

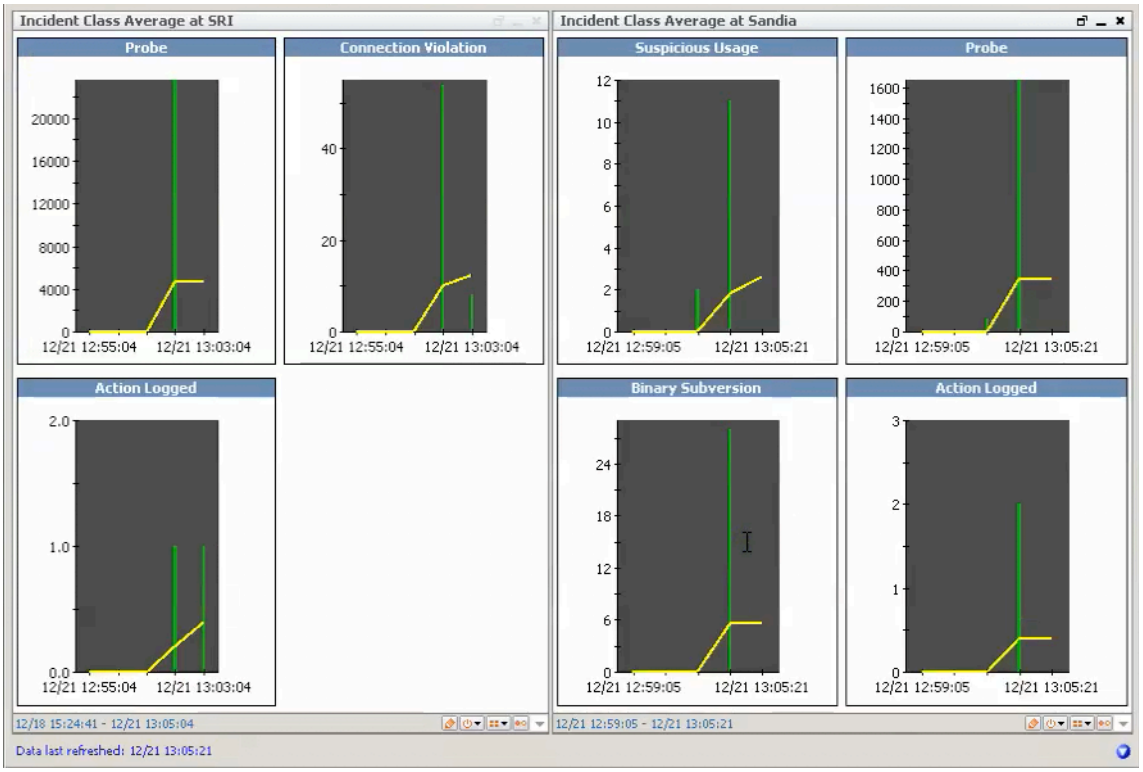
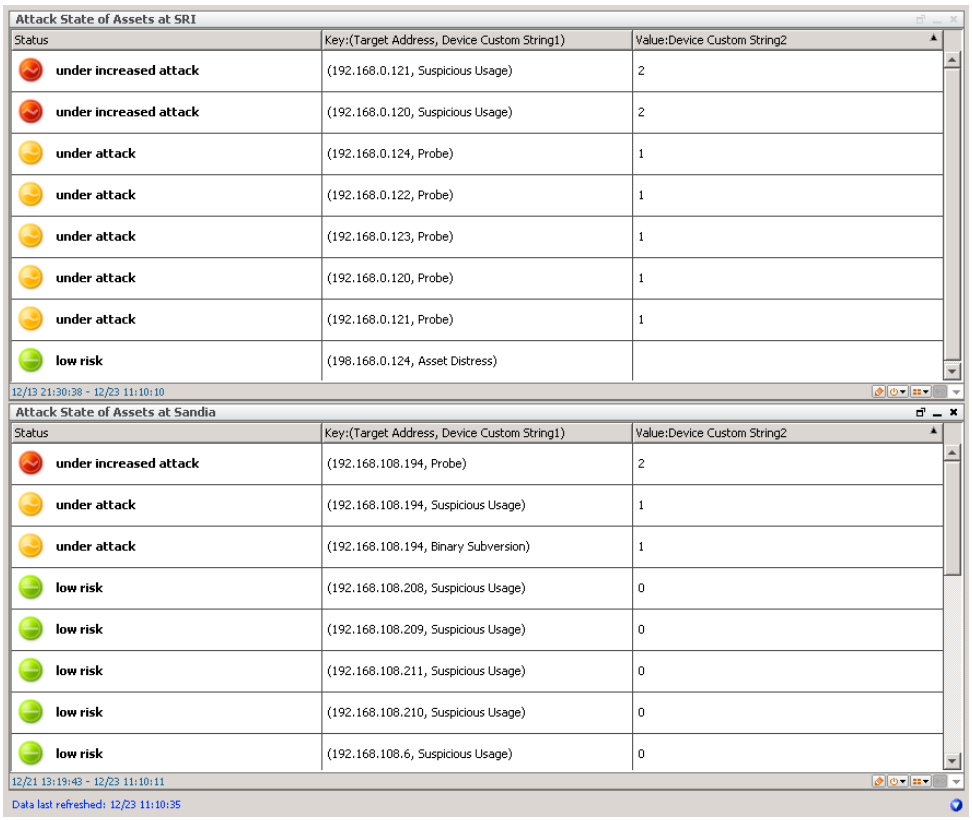


Figure 19. Data monitor statistics panel

Once the threshold criterion for an incident class is met, we declare this class to be "increased". In a different ArcSight implementation, we are keeping track of all IP addresses that are inside the modeled networks. Then, the list of IP addresses that have been seen as targets in the event stream are visualized similar to Figure 20. Here, the status of each pair of <IP address, Incident Class> is colored according to the state transition diagram introduced in the previous section. A red symbol means that this IP address has seen this incident class and this incident class is currently coming in with an increased rate. Yellow symbols denote events that pertain to incident classes that are currently not labeled "increased". Green symbols mean that we have not seen an event of this incident class for this IP address for a certain time.



Attack State of Assets at SRI

Status	Key:(Target Address, Device Custom String1)	Value:Device Custom String2
under increased attack	(192.168.0.121, Suspicious Usage)	2
under increased attack	(192.168.0.120, Suspicious Usage)	2
under attack	(192.168.0.124, Probe)	1
under attack	(192.168.0.122, Probe)	1
under attack	(192.168.0.123, Probe)	1
under attack	(192.168.0.120, Probe)	1
under attack	(192.168.0.121, Probe)	1
low risk	(198.168.0.124, Asset Distress)	

12/13 21:30:38 - 12/23 11:10:10

Attack State of Assets at Sandia

Status	Key:(Target Address, Device Custom String1)	Value:Device Custom String2
under increased attack	(192.168.108.194, Probe)	2
under attack	(192.168.108.194, Suspicious Usage)	1
under attack	(192.168.108.194, Binary Subversion)	1
low risk	(192.168.108.208, Suspicious Usage)	0
low risk	(192.168.108.209, Suspicious Usage)	0
low risk	(192.168.108.211, Suspicious Usage)	0
low risk	(192.168.108.210, Suspicious Usage)	0
low risk	(192.168.108.6, Suspicious Usage)	0

12/21 13:19:43 - 12/23 11:10:11

Data last refreshed: 12/23 11:10:35

Figure 20. Data monitor statistics table view

Figure 21 shows two event graphs for potential DDoS attacks at each modeled utility. Red squares denote source IP addresses of events that have a current DDoS target as the destination. Each blue square denotes the target IP address currently suspected under attack. As we have modeled the networks of our testbed utilities in ArcSight, we also display the asset name for each IP address in these graphs. Each arrow between a source and destination address is labeled with the incident class of the corresponding events. The number of events translates into different sizes of light blue circles. We emphasize Asset Distress events of the suspected target IP by forming a loop in these event graphs. Once human operators watching how these event graphs develop see such a loop, they can be more certain of witnessing an actual DDoS attack.

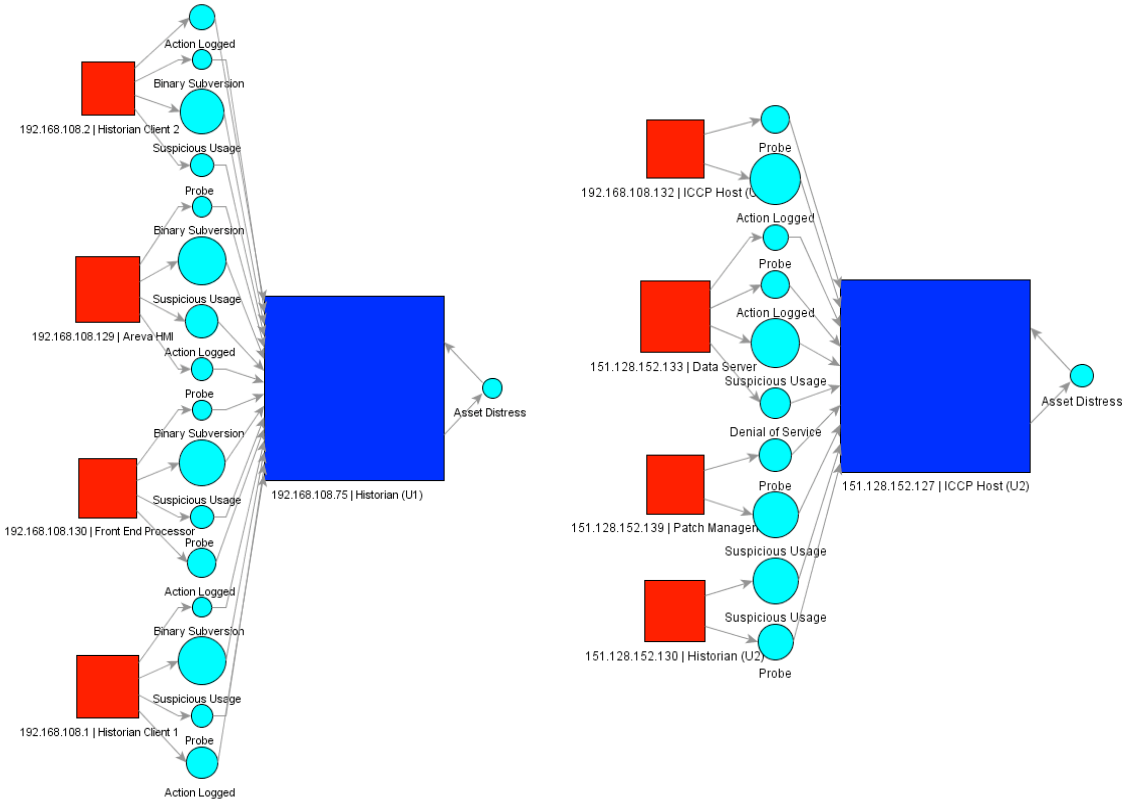


Figure 21. Visualization of DDoS attack

For the network traversal attack, once a sequence of events causes the chain rules in ArcSight to fire, we collect all endpoints of these events as pairs in a so-called Active List. Then, all subsequent events that have endpoints that match any of the pairs in the list are collected and visualized, building a comprehensive picture of the ongoing network traversal attack.

Our visualization as shown in Figure 22 uses the ArcSight event graph data monitor to draw each endpoint as a square and events as circles. To show how the network traversal attack is indeed penetrating the layers to reach its ultimate goals of highly critical field devices and is even crossing into a different utility network, we identify the zones to which the hosts belong.

Red squares are sources of events, blue squares act as both sources and destinations, and white squares are destinations. The squares are labeled with the host name of the IP addresses drawn from the ArcSight network model of the two utilities we simulated in our testbeds. Also included in the label is the location of the host. U1 refers to Utility 1 and U2 to Utility 2.

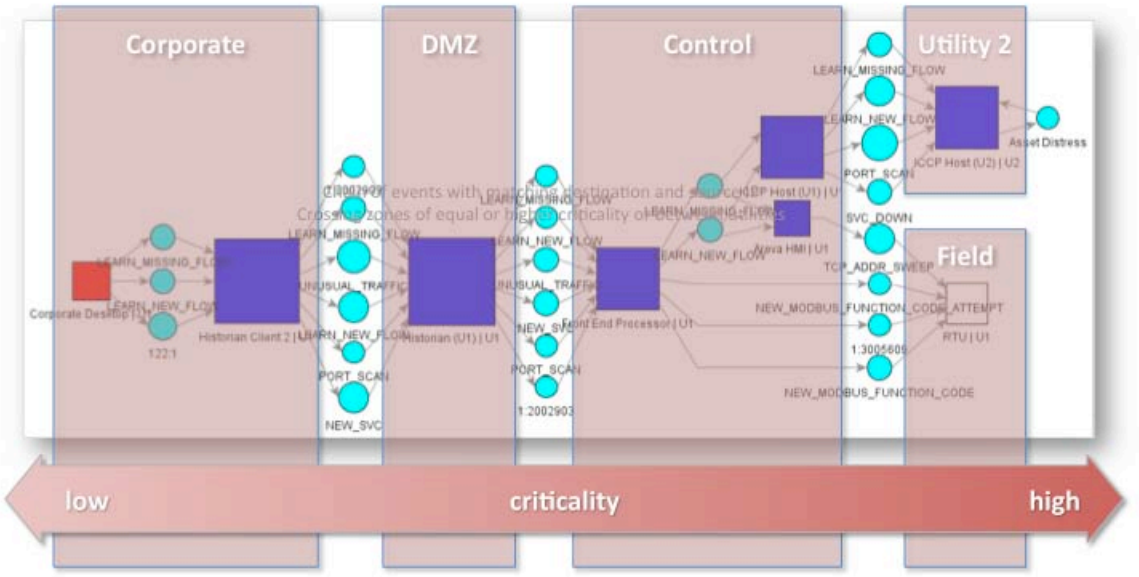


Figure 22. Visualization of network traversal

Each event circle has a size that is proportional to the number of events observed for that event type, and the label below each circle shows the event name. The event names shown in Figure 22 reflect the diversity of the underlying detection components contributing to the visualization. Events relating to new or missing flows are generated by the flow anomaly detection component. Numeric labels correspond to Snort identifiers; for example, the label 1:2002903 denotes a Snort rule for detecting a specified shellcode segment, part of an exploit against Wintel machines. The visualization illustrates the various scans and exploits used by the attackers as they traverse from the corporate network through the DMZ and into the control network of Utility 1, from which they attempt to compromise a field asset and also attack Utility 2 via the communication channel that is used for ICCP. As shown, the figure depicts the result of the complete traversal attack; the various nodes and edges are dynamically generated as ArcSight receives events matching the correlation rules, and the operator can watch the attack graph build up from left to right as the attack progresses.[1]

As the events composing this illustration of a network traversal attack contain much more information than shown in the event graph, we have also implemented a table view of the events below the graph view in a customized dashboard, in which the operator can quickly look up more details of each event such as the time stamp, a more detailed message, and the incident class. Finally, ArcSight provides more means of notifying operators—for example, through email or paging. It would be straightforward to implement a rule with the specific parameters for a utility

[1] Refer to http://www.csl.sri.com/projects/dates/distributech.html for a video of a DATES demo.

employing this system to send out automated alerts when the chain has reached a certain length, or certain zones or other important criteria are met.

Outreach

DATES project staff members have been active in industry forums, conferences, roadmap workshops, and other outreach. The following are some events at which we presented project results. Conference papers from the proceedings of HICSS and IEEE HST are referenced at the end of this report.

- SANS SCADA Summit, New Orleans, Louisiana, 2008, introduction of the DATES project to a panel chaired by Hank Kenchington of the Department of Energy.
- ESec-Northwest, Portland, Oregon, March 2008, presentation of DATES to representatives from several utilities in the Northwest, seeking feedback and participation.
- DOE Roadmap Workshops, May 2008 (Chicago, Illinois) and September 2009 (San Diego, California), discussion participation.
- Trustworthy Cyber Infrastructure for the Power Grid (TCIP), Lake Geneva, Wisconsin, June 2008, DATES overview as part of a week-long summer school in cyber security for infrastructure systems.
- Process Control Systems Forum, San Diego, California, August 2008, further results on flow anomaly detection.
- EPRI SmartGrid Workshop, Denver, Colorado, September 2008, general project brief.
- Hawaii International Conference on System Sciences, January 2009, further results on flow anomaly detection.
- IEEE Workshop on Homeland Security Technologies (IEEE-HST), May 2009, further results on flow anomaly detection.
- ArcSight Users Group Meeting, Potomac, Maryland, panel participation.
- EnergySec, Seattle, Washington, September 2009, overall project brief.
- TCIP, Urbana, Illinois, November 2009, panel participation.
- DistribuTech, Tampa, Florida, March 2010, integrated demonstration of detection and security event correlation (see http://www.csl.sri.com/projects/dates/distributech.html).

Summary

The energy sector increasingly relies on digital industrial control systems (ICS) such as Supervisory Control and Data Acquisition (SCADA) to operate complex cyber-physical systems. Legacy control systems were isolated and used proprietary protocols, achieving a degree of "security through obscurity". Modern systems increasingly use open standards such as Internet protocols, and are increasingly interconnected. Although this has resulted in improved safety and cost-effectiveness of operation, there is concern that these systems are vulnerable to cyber attacks similar to those that have long affected enterprise systems. In control systems, there is the additional concern that successful attacks might cause not merely economic loss, but possibly environmental and safety impacts as well. The concern is sufficiently serious that the Department of Energy has engaged a panel of experts to draft a roadmap to secure control systems in the energy sector, and to continually update this roadmap to comprehend new threats as well as technological advances.

The DATES (Detection and Analysis of Threats to the Energy Sector) project has developed a distributed, multi-algorithm intrusion detection capability suitable for the digital control systems that operate much of our energy infrastructure. The detection capability combined conventional IDS signature approaches with novel components using Bayesian methods and learning-based anomaly detection. These latter components were shown to be effective in control system environments due to the regularity of traffic and limited number of protocols in these environments. The detection capability was integrated with a leading Security Information/Event Management (SIEM) capability from ArcSight. The integrated detection and SIEM solution provides a level of situational awareness for a variety of attacks against control systems, in support of DOE Cybersecurity Roadmap objectives in monitoring.

The DATES solution extends the state of the practice in ICS monitoring by implementing the following:

- Multiple detection algorithms, including an ICS-aware Snort knowledge base, as well as SRI's components for stateful packet inspection, probabilistic/Bayesian analysis, and event threading.
- Unique model-based detection capability, including a communication pattern anomaly detection module, which leverages the unique traffic characteristics of ICS to facilitate detection of novel attacks such as zero-day exploits.
- Non-intrusive network monitoring design based on passive listening and employing a separate network interface for event reporting. This makes the monitoring appliance invisible to conventional network scans, and guarantees that the critical function of the ICS is not affected at all.
- DATES monitoring components interface with the advanced market-leading ArcSight SIEM Platform, and can easily be adapted to communicate with other types of event-consuming components.

- Alert correlation heuristics are designed for ICS environments. For example, because process control networks typically have constrained communication patterns enforced by network firewalls, and the high-valued assets such as RTUs are not directly accessible from comparatively less secure network zones (e.g., business network zone), we developed techniques to detect and correlate alerts corresponding to network traversal attacks.

DATES was sponsored by the United States Department of Energy and performed by a team led by SRI International, with collaboration from Sandia National Laboratories, ArcSight, Inc., and Invensys Process Systems. The project ran from October 2007 through March 2010. As components were developed, they were evaluated in test environments operated in parallel at SRI and Sandia. The SRI system was based on a commercial system from Invensys, a leading vendor of industrial control systems. Sandia implemented a Virtual Control System Environment (VCSE), which allows flexible inter-operation of virtual and physical digital control components. Over the last six months of the project, these environments were used for a series of cross-site experiment scenarios of increasing complexity. A demonstration based on the findings of these cross-site experiments was presented at the Distributech Conference in March 2010.

The DATES team engaged in extensive outreach activity, with presentations at conferences and industry forums. Although there was interest in the project, we were unfortunately not able to implement a pilot deployment at an asset owner facility.

Acknowledgement

This material is based upon work supported by the Department of Energy under Award Number DE-FC26-07NT43314.

References

[Bayes] Alfonso Valdes and Keith Skinner. Adaptive, Model-based Monitoring for Cyber Attack Detection, from *Recent Advances in Intrusion Detection (RAID 2000)*, Springer-Verlag Lecture Notes in Computer Science (LNCS 1907).

[DB2010] Dale Peterson et al., http://www.digitalbond.com/index.php/research/, last accessed March 23, 2010.

[DNP3] The DNP Users Group, http://www.dnp.org, last accessed March 25, 2010.

[DOE2006] Jack Eisenhauer, Paget Donnelly, Mark Ellis, and Michael O'Brien, *Roadmap to Secure Control Systems in the Energy Sector*, prepared for U. S. Department of Energy, U. S. Department of Homeland Security, and Natural Resources Canada.

[EMERALD] Philip Porras and Peter Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, *Proc. 1997 Network Information Security Conference*.

[EMERG] OISF Emerging Threats, http://www.emergingthreats.net/, last accessed March 23, 2010.

[HST 09] Alfonso Valdes and Steven Cheung, Communication Pattern Anomaly Detection in Process Control Systems, *Proc. 2009 IEEE International Conference on Technologies for Homeland Security,* Waltham, MA, May 11-12, 2009.

[ICCP] International Electrotechnical Commission, IEC 60870-6/TASE 2, http://www.iec.ch, last accessed March 25, 2010.

[Invensys] Invensys Process Systems, http://www.ips.invensys.com/en/products/autocontrols/Pages/DistributedControl-IASeries-P018.aspx, last accessed March 23, 2010.

[LOGIIC] Linking the Oil and Gas Industry to Improve Cybersecurity, http://www.cyber.st.dhs.gov/logiic.html, last accessed March 23, 2010.

[McAfee] McAfee AntiVirus Enterprise, http://www.mcafee.com/us/enterprise/products/system_security/servers/virusscan_enterprise.html, last accessed March 30, 2010.

[MCorr] Phillip Porras, Martin Fong, and Alfonso Valdes, A Mission-Impact-Based Approach to INFOSEC Alarm Correlation, *Proc. Recent Advances in Intrusion Detection,* Zurich, Switzerland, October 2002.

[Modbus] The Modbus Organization, http://www.modbus.org, last accessed March 25, 2010.

[S4] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes, Using Model-based Intrusion Detection for SCADA Networks, *Proc. SCADA Security Scientific Symposium,* Miami Beach, Florida, January 2007.

[SEP] Symantec Endpoint Protection system, http://www.symantec.com/business/endpoint-protection, last accessed March 25, 2010.

[SNL2010] Michael J McDonald, John Mulder, Bryan T Richardson, Regis H. Cassidy, Adrian Chavez, Nicholas D Pattengale, Guylaine M Pollock, Jorge Mario Urrea, Moses Daniel Schwartz, William Dee Atkins, and Ronald D. Halbgewachs, *Modeling and Simulation for Cyber-Physical System Security Research, Development and Applications,* Sandia Report SAND2010-0568, February 2010

[SNORT] SNORT Open Source Intrusion Detection, http://www.snort.org/, last accessed March 23, 2010.

DATES Publications

[HICSS 09] Alfonso Valdes and Steven Cheung, Intrusion Monitoring in Process Control Systems, *Proc. 42nd Hawaii International Conference on System Sciences,* Big Island, Hawaii, January 5-8, 2009.

[HST 09] Alfonso Valdes and Steven Cheung, Communication Pattern Anomaly Detection in Process Control Systems, *Proc. 2009 IEEE International Conference on Technologies for Homeland Security,* Waltham, MA, May 11-12, 2009.

[PST 10] Linda Briesemeister, Steven Cheung, Ulf Lindqvist, and Alfonso Valdes, Detection, Correlation, and Visualization of Attacks Against Critical Infrastructure Systems, in submission to the *Eighth Annual Conference on Privacy, Security and Trust,* August 2010.