Detection, Correlation, and Visualization of Attacks Against Critical Infrastructure Systems

Linda Briesemeister, Steven Cheung, Ulf Lindqvist, Alfonso Valdes
SRI International, Menlo Park, CA
firstname.lastname@sri.com

***Abstract*—Digital control systems are essential to the safe and efficient operation of a variety of industrial processes in sectors such as electric power, oil and gas, water treatment, and manufacturing. Modern control systems are increasingly connected to other control systems as well as to corporate systems. They are also increasingly adopting networking technology and system and application software from conventional enterprise systems. These trends can make control systems vulnerable to cyber attack, which in the case of control systems may impact physical processes causing environmental harm or injury.**

We present some results of the DATES (Detection and Analysis of Threats to the Energy Sector) project, wherein we adapted and developed several intrusion detection technologies for control systems. The suite of detection technologies was integrated and connected to a commercial security event correlation framework from ArcSight. We demonstrated the efficacy of our detection and correlation solution on two coupled testbed environments. We particularly focused on detection, correlation, and visualization of a network traversal attack, where an attacker penetrates successive network layers to compromise critical assets that directly control the underlying process. Such an attack is of particular concern in the layered architectures typical of control system implementations.

***Index Terms*—critical infrastructure security, control system security, intrusion and anomaly detection, alert correlation, security information event management**

I. Introduction

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 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 combines conventional 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 because of the regularity of traffic and limited number of protocols in these environments [1]. We integrate the detection capability with a leading Security Information Event Management (SIEM) system from ArcSight, which overall provides a monitoring solution that complements perimeter defenses and provides the ICS security operator a significantly improved level of situational awareness for a variety of attacks against control systems.

In this paper, we present our integrated system using the example of a so-called *network traversal attack*. Such attacks are of particular importance in infrastructure systems because of the layered architectures of such systems. A typical control system is isolated from public and corporate networks by demilitarized zones (DMZs) and other network segmentation architectures. The controlled connections between the networks reflect operational requirements. Therefore, an attack that penetrates network layers in succession exploits trust relationships between networks, providing a traversal path for an attacker from a public network all the way to high-priority field devices.

To define network traversal attacks, we assume that each host H has a criticality score, denoted by $criticality(H)$, which may depend on its functionality or the value of the data it manages. We model network traversal attacks as sequences of network connections of the form $H_1 \rightarrow H_2 \rightarrow \cdots \rightarrow H_k$ such that H_i are hosts and $criticality(H_{i-1}) \leq criticality(H_i)$, where $i \in \{2, \cdots, k\}$. Moreover, we assume that each connection, $H_{i-1} \rightarrow H_i$, in the sequence pertains to one or more events that violate a security policy, for example, an event that violates a network access policy.

Our definition of network traversal attacks is similar to that of stepping-stone attacks (e.g., [2], [3]), but they differ in their motivation. For stepping-stone attacks, an adversary uses multiple intermediate hosts to obfuscate the origin of the attack. In network traversal attacks, an adversary exploits the trust relationships among hosts to attack a high-value target host to which the adversary does not have direct access.

II. System Architecture

We consider a corporate or enterprise network (possibly Internet-facing), with clients that access resources such as historian servers in a DMZ between corporate and control zones. These historian servers receive data from field control processors (FCPs) or front end processors (FEPs) in the

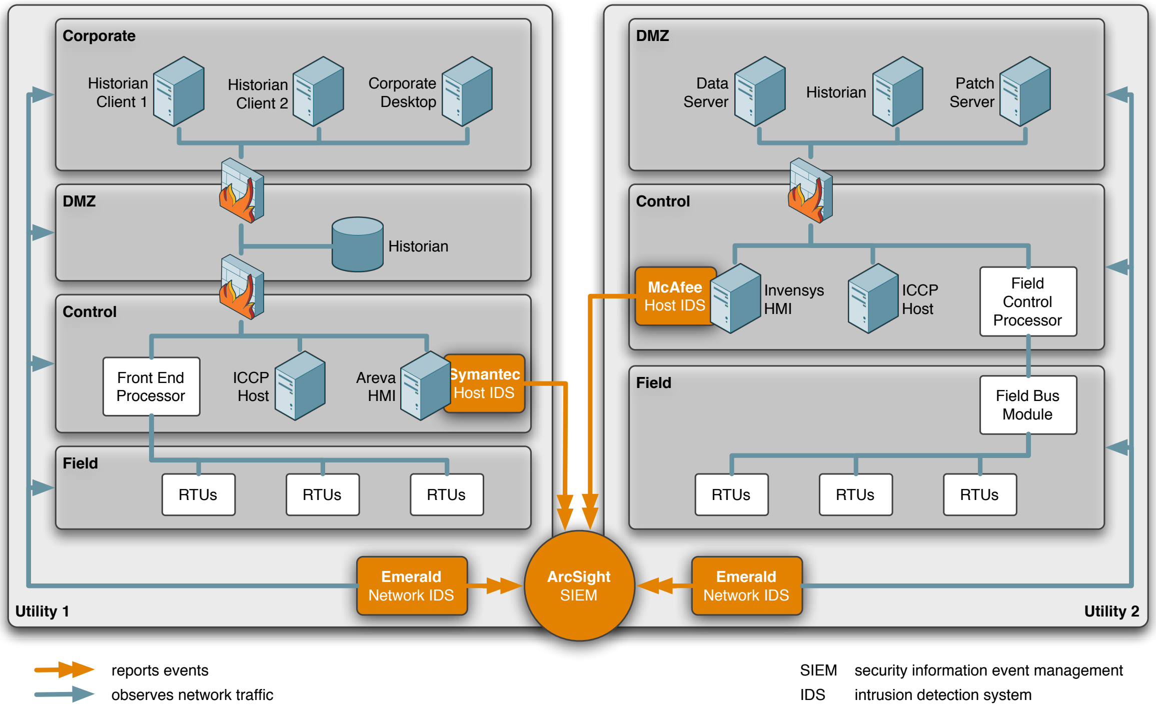


Fig. 1. Testbed architecture overview

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 computer platforms. The field network devices directly monitor and control a physical process, such as refining, manufacturing, or electric power generation/transmission/distribution. Separation of enterprise and control networks via a double-firewalled DMZ is consistent with good practices widely used in control systems [4].

The control 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 through the firewall 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 testbed architectures as shown in Figure 1 are instantiations of this reference architecture, with logical separation of corporate, DMZ, control, and field networks.

A. Intrusion Detection for Control Systems

Because process control systems typically consist of enterprise commercial off-the-shelf (COTS) components (e.g., commercial database systems running on Microsoft Windows) and process-control-specific components, such as remote terminal units (RTUs) that communicate using Modbus TCP, our intrusion monitoring approach employs a suite of intrusion detection sensors for both enterprise and process-control-specific subsystems to achieve good attack detection coverage.

Intrusion Detection for Enterprise Networks and Hosts: To monitor enterprise networks (such as the corporate network in the reference architecture as shown in Figure 1) and COTS components (such as commercial database systems and operating systems), we employ intrusion detection sensors monitoring events at the network and host levels.

For network monitoring, we employ Snort [5] for attack-signature-based detection, Bayesian sensors for detecting several important attack classes such as reconnaissance and asset distress [6], and EMERALD eXpert sensors for performing deep packet inspection and stateful analysis for several key network protocols such as HTTP [7], [8].

For host monitoring, we employ commercial security solutions (in particular, Symantec Endpoint Protection [9] and McAfee VirusScan Enterprise [10]) 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 Subsystem: Our control-system-specific monitoring solution employs both the signature-based approach and the model-based detection approach for monitoring control system environments and PCS protocols, e.g., Modbus, DNP3, and the Inter Control Center Protocol (ICCP).

For the signature-based approach, we use Snort [5] with the control-network-specific rules developed by Digital Bond [11]. This detection capability provides high-fidelity detection for known malicious activities.

In model-based detection, we characterize the expected behavior of the system, and detect attacks when the system deviates from this behavior. This approach has the potential to detect unknown attacks, and may provide complementary detection coverage to the signature-based approach. We note that control systems typically exhibit regular and predictable communication patterns, which significantly simplifies the specification or learning of these models.

We have developed several model-based monitors for process control systems, including eModbus for detecting changes in server or service availability for Modbus servers, eFlowmon for performing flow-based anomaly detection for monitoring the traffic patterns for individual network flows, and Snort rule sets for detecting violations of the Modbus protocol specification. Readers are referred to [1], [12] for detailed description of our work on model-based detection.

B. Event Management

Control systems monitor physical 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.

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.

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. We employ the incident classification developed in our previous work on alert correlation [13].

TABLE I
INCIDENT CLASSES AND THEIR PRIORITIZATION

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

TABLE II
ASSET TYPE CRITICALITIES

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

Based on the relative importance among the security objectives for process control systems, we developed a prioritization scheme for incident classes. Generally, asset owners consider availability as the most important security objective, followed by integrity, and then confidentiality. We then 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 “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. Table I shows the incident classes with their prioritization.

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 and network zones to reflect their criticality. Like incident classes, asset type and network zone present a high-level abstraction that enables us to specify

TABLE III
NETWORK ZONE CRITICALITIES

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

general correlation criteria for entire classes of assets as opposed to those for specific asset instances. Tables II and III show the asset types and network zones with their associated criticality values.

III. TESTBEDS

To study cross-site attack detection and correlation, we employ two testbeds for experimentation and validation. These two testbeds, hosted at SRI International and Sandia National Laboratories, respectively, each model a utility company in the electric industry. Moreover, during the experiments, the two testbeds were linked using a secure connection so that their alerts could be securely sent to the same ArcSight SIEM server for event correlation. Figure 1 depicts a high-level schematic for the testbeds.

The SRI testbed is based on a Distributed Control System (DCS) from Invensys Process Systems, IA series [14], and features the following key components:

- An application workstation (AW) for configuration, visualization, and control
- A control LAN based on a redundant pair of Enterasys switches (optical Ethernet)
- An Invensys field control processor (FCP) module
- A field bus that connects the FCP to two Ethernet field bus modules (FBMs)
- A field LAN connecting the FBM and simulated Modbus devices (Modbus simulators from Modbustools.com and Calta) running in virtual machines
- Network and host intrusion detection sensors, configured to send security events to the ArcSight SIEM server

The Sandia testbed is based on the Virtual Control System Environment (VCSE) [15], a flexible, distributed tool-oriented environment enabling real, emulated, and simulated components to be brought together to facilitate the analysis of the impact of threats against cyber-physical systems.

Both testbeds are running an instantiation of SRI's EMERALD system, which combines the aforementioned network intrusion detection technologies. We modified EMERALD to create output in ArcSight's Common Event Format (CEF), which ArcSight obtains using its SmartConnector technology. In CEF, we transmit the source and destination IP addresses and ports, as well as the name, description, and incident class (using the field "Device Event Category") of alerts. We configured the host-based intrusion detection systems such as Symantec and McAfee to generate messages in syslog format, which ArcSight picks up using its syslog SmartConnector.

In this work, researchers at Sandia and SRI instrumented an instantiation of VCSE to develop and test control system intrusion detection, SIEM, and large-scale threat analysis technologies. Moreover, we employed 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 and a variety of attack scenarios.

IV. ATTACK SCENARIO

SRI and Sandia conducted a number of attack scenarios against the two test environments, including one attack scenario to simulate an attack from a compromised utility to another utility using ICCP [16].

One attack scenario consisted of a distributed denial of service using multiple compromised computers inside utility networks to attack a single target. The DATES framework detected the various exploits used, and the anomaly detection components alerted when the attack successfully impeded expected communication flows.

Another attack scenario, which we describe in further detail here, was a network traversal attack where the attacker successively penetrated the defensive layers of the control system architecture. The network traversal is a multi-step attack using a number of scans and exploits that take advantage of common vulnerabilities as well as specific attacks against control system protocols and assets. The attacker uses a strategy of compromising progressively more critical nodes, seeing what can be attacked from that point, and proceeding to the next target.

In our scenario, the adversary has compromised a corporate desktop computer at Utility 1. From this computer, the adversary gains a foothold on a historian client, still in the corporate zone. This client is permitted to access the historian server in the DMZ. The attacker finds a vulnerability on the historian server, compromises it, and uses it as a stepping stone to attack the front end processor in the control network. From this node, the attacker unsuccessfully attempts to trip a breaker on an RTU in the field network, and also attacks the HMI. Although these attacks are unsuccessful, they are detected through network and host protection components. The attacker compromises the ICCP host in Utility 1, which is permitted to communicate to the corresponding ICCP host at Utility 2. By exploiting this communication channel, the adversary is able to bring down the ICCP host at Utility 2.

V. CORRELATION TO EXPOSE NETWORK TRAVERSAL

We detect and visualize network traversal attacks in the ArcSight SIEM component to identify a certain pattern of events. Thus, the flood of low-level events, which ArcSight processes and stores, and which may contain false positives, is abstracted into a case that a human operator can easily grasp given our visualization (described in more detail in Section VI).

In our attack scenario, we correlate events that form a chain and 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, 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 (see Table III). As shown in the algorithm below, 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 event.

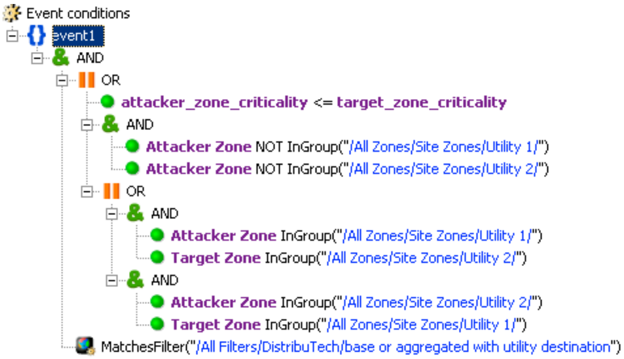
Given a set of alerts A_1, A_2, A_3, etc., the zone-based criticality escalation algorithm will correlate them as an event chain if the following conditions are met:

(1) zone(dst(A_i)) is "internal"
(2) dst(A_i) = src(A_{i+1})
(3) criticality(zone(src(A_i))) $\leq$ criticality(zone(dst(A_i)))
OR zone(src(A_i)) is "external"
OR utility(zone(src(A_i))) $\neq$ utility(zone(dst(A_i)))

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.

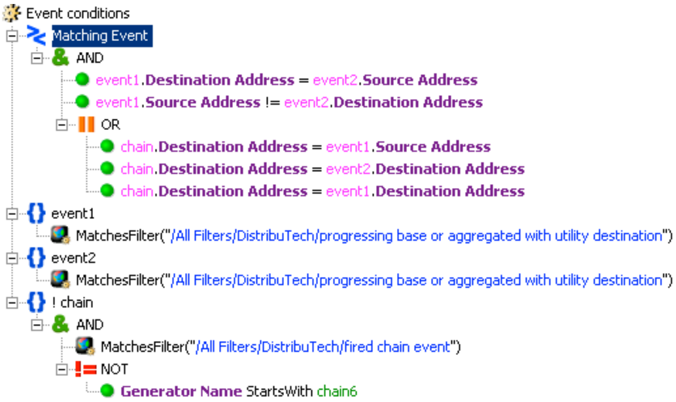
Condition (1) establishes the boundary case that the event chaining process stops when the destination zone of the last event is no longer being monitored. Condition (2) corresponds to the requirement about matching the destination IP address of an event with the source IP address of another one. Condition (3) pertains to the criterion about nondecreasing zone criticality (i.e., the zone that corresponds to the destination IP address of an event should be at least as critical as that which corresponds 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.

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 2 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 the one of the destination zone or, b) the source is outside of the utility networks, or c) the event crosses from one to the other utility.



```
Event conditions
  event1
    AND
      OR
        attacker_zone_criticality <= target_zone_criticality
        AND
          Attacker Zone NOT InGroup("/All Zones/Site Zones/Utility 1/")
          Attacker Zone NOT InGroup("/All Zones/Site Zones/Utility 2/")
        OR
          AND
            Attacker Zone InGroup("/All Zones/Site Zones/Utility 1/")
            Target Zone InGroup("/All Zones/Site Zones/Utility 2/")
          AND
            Attacker Zone InGroup("/All Zones/Site Zones/Utility 2/")
            Target Zone InGroup("/All Zones/Site Zones/Utility 1/")
      MatchesFilter("/All Filters/DistribuTech/base or aggregated with utility destination")
```

Fig. 2. ArcSight definition of filter for potential traversal events



```
Event conditions
  Matching Event
    AND
      event1.Destination Address = event2.Source Address
      event1.Source Address != event2.Destination Address
      OR
        chain.Destination Address = event1.Source Address
        chain.Destination Address = event2.Destination Address
        chain.Destination Address = event1.Destination Address
  event1
    MatchesFilter("/All Filters/DistribuTech/progressing base or aggregated with utility destination")
  event2
    MatchesFilter("/All Filters/DistribuTech/progressing base or aggregated with utility destination")
  ! chain
    AND
      MatchesFilter("/All Filters/DistribuTech/fired chain event")
      NOT
        Generator Name StartsWith chain6
```

Fig. 3. ArcSight definition of rule "chain2"

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 ith 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 firing rule 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 make heavy but careful use of this feature 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 3 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,

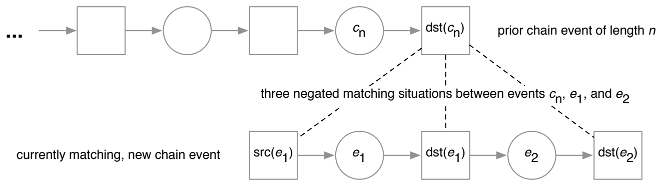
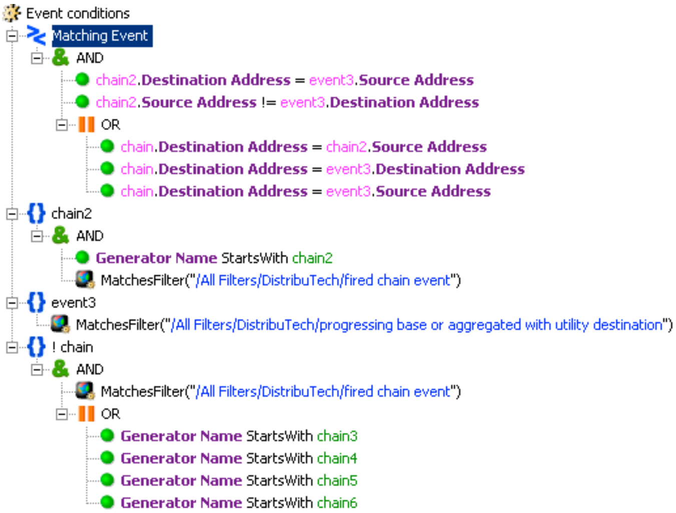

...
c_n
dst(c_n)
prior chain event of length n
three negated matching situations between events c_n, e_1, and e_2
currently matching, new chain event
src(e_1)
e_1
dst(e_1)
e_2
dst(e_2)

Fig. 4. Three negated matching situations for "chain2"



```
Event conditions
  Matching Event
    AND
      chain2.Destination Address = event3.Source Address
      chain2.Source Address != event3.Destination Address
      OR
        chain.Destination Address = chain2.Source Address
        chain.Destination Address = event3.Destination Address
        chain.Destination Address = event3.Source Address
  chain2
    AND
      Generator Name StartsWith chain2
      MatchesFilter("/All Filters/DistribuTech/fired chain event")
  event3
    MatchesFilter("/All Filters/DistribuTech/progressing base or aggregated with utility destination")
  ! chain
    AND
      MatchesFilter("/All Filters/DistribuTech/fired chain event")
      OR
        Generator Name StartsWith chain3
        Generator Name StartsWith chain4
        Generator Name StartsWith chain5
        Generator Name StartsWith chain6
```

Fig. 5. ArcSight definition of rule "chain3"

i.e., $\text{dst}(e_1) = \text{src}(e_2)$ and $\text{src}(e_1) \neq \text{dst}(e_2)$. Both events must match the filter "progressing base or aggregated with utility destination" defined in Figure 2.

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 4. 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 $\text{src}(e_1)$, $\text{dst}(e_1)$, and $\text{dst}(e_2)$, 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.

We then define the rules "chain3" through "chain6" as follows and will use "chain3" as shown in Figure 5 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., $\text{dst}(c_2) = \text{src}(e_3)$ and $\text{src}(c_2) \neq \text{dst}(e_3)$. 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 4.

VI. Visualization

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 6 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 even crosses into a different utility network, we identify the zones to which the hosts belong in the picture.

Red (medium gray) squares are sources of events, blue (dark gray) squares act as both sources and destinations, and white squares are destinations of events. 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.

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 6 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. See Section IV for a description of the network traversal attack. 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

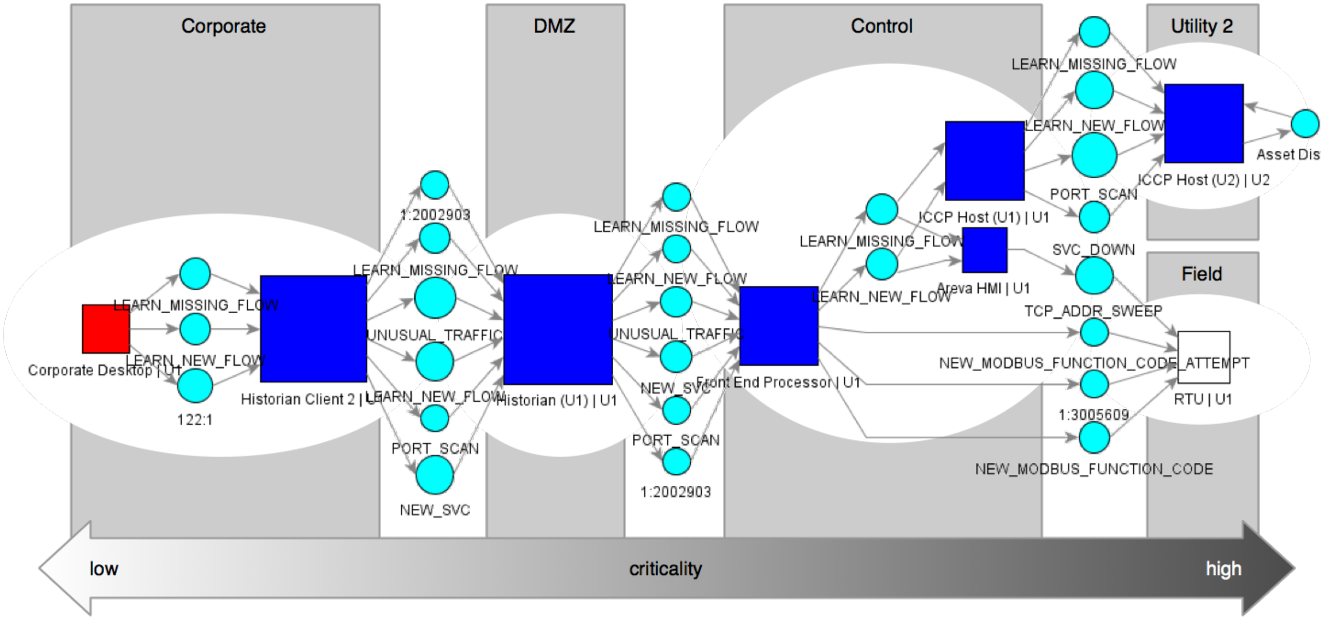


Fig. 6. Visualization of network traversal attack in ArcSight

right as the attack progresses (refer to [17] for a video of a DATES demo).

As the events comprising this picture 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 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.

VII. Related Work

The ubiquitous interconnection of industrial control systems and the migration of these systems to commodity platforms such as Windows for HMI as well as TCP/IP networking and embedded operating systems in field devices have led to intense recent interest in cyber security for these environments. The Department of Energy has led the development of an evolving industry roadmap document to advance cyber security in energy sector control systems [18]. This roadmap identifies monitoring as an important support for the goal to develop control systems that are secure and resilient to attack.

Numerous agencies have developed best-practice architectures for connecting control systems to enterprise systems [4], [19]. Sandia's VCSE, which comprises part of the DATES testbed, instantiates a variety of control architectures in a mix of virtual and physical components [15].

With respect to intrusion detection in control systems, Digital Bond developed a set of Snort signatures tailored to detect attacks against several important control protocols, and it has continued to maintain this signature base [11], [5]. Our DATES detection suite includes this signature base among its other algorithmic components.

The efficacy of anomaly detection systems that take advantage of the regularity of traffic in control systems was demonstrated by Cheung et al. [1]. DATES incorporates these concepts and extends them to flow anomaly detection, and also introduces an adaptive learning feature.

The DHS LOGIIC (Linking the Oil and Gas Industry to Improve Cybersecurity) Correlation Project was an early demonstration of integrated detection and SIEM in a control system environment, and was a starting point for the DATES solution [20]. DATES developed advances in anomaly detection, cross-site visualization, and visualization of zone traversal attacks.

From the industry side, commercial solutions for intrusion detection and prevention are available from a variety of sources, with leading products from Tofino and Industrial Defender, among others [21], [22]. Vendor systems often include commercial security components as well; for example, the Invensys IA system in our testbed includes a McAfee host IDS [14], [10].

VIII. Conclusion

Although digital control systems are adopting many aspects of conventional enterprise computing, they differ from enterprise systems in several key respects. The need for emergency operator intervention and for continuous operation makes many enterprise security practices difficult to apply. On the other hand, the mission of control systems is much narrower in scope than that of enterprise systems. Control systems typically run a small set of comparatively simple protocols, and exhibit regular communication patterns between assets in various network zones. This regularity makes anomaly detection approaches more effective in terms of detection sensitivity as well as false alarm rate than in enterprise systems.

The layered defense makes network traversal attacks particularly important to detect. Since by their very nature these attacks use multiple exploits to compromise assets closer and closer to the highest-value field components, a correlation and visualization framework must be imposed on the intrusion detection system to provide meaningful situational awareness.

We have presented results from the DATES project, in which we combined a variety of detection approaches, including new approaches using learning-based anomaly detection, along with the ArcSight Security Information Event Management system, to provide a situational awareness solution suitable for control systems such as those widely used in the energy sector. We demonstrated the suite of components on two interconnected testbeds. The project team ran a number of cross-site and network traversal attacks against the testbed. These attacks were detected by a combination of conventional signature techniques as well as by the anomaly detection techniques developed for DATES. The SIEM provided a particularly rich visualization of a network traversal attack.

Acknowledgment and Disclaimer

We thank our project partner Sandia National Laboratories, and especially Regis Cassidy for standing up the testbed and creating attacks to perform red teaming of our system. Martin Fong and Hal Huntley at SRI provided exceptional help in the implementation and running of our testbeds. We also thank ArcSight and the ArcSight user community members who have provided help while we were implementing our system.

This paper is based upon work supported by the United States Department of Energy under Award Number DE-FC26-07NT43314. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proceedings of the SCADA Security Scientific Symposium*, Miami Beach, Florida, Jan. 2007.

[2] A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping stones: Algorithms and confidence bounds," in *Recent Advances in Intrusion Detection (RAID)*. Springer, 2004, pp. 258–277.

[3] S. Staniford-Chen and L. Heberlein, "Holding intruders accountable on the Internet," in *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, 1995, pp. 39–49.

[4] British Columbia Institute of Technology (BCIT), "NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks. Revision 14," Feb. 2005.

[5] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proceedings of LISA '99: 13th Systems Administration Conference*, Seattle, Washington, Nov. 7–12, 1999, pp. 229–238.

[6] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *Recent Advances in Intrusion Detection (RAID 2000)*, ser. LNCS, H. Debar, L. Me, and F. Wu, Eds., Toulouse, France, Oct. 2000.

[7] P. Porras and P. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," in *Network Information Security Conference*, 1997.

[8] U. Lindqvist and P. A. Porras, "Detecting computer and network misuse through the production-based expert system toolset (P-BEST)," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, Oakland, California, May 9–12, 1999, pp. 146–161.

[9] "Symantec Endpoint Protection System," last accessed March 25, 2010. [Online]. Available: http://www.symantec.com/business/endpoint-protection

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

[11] Digital Bond, "IDS signatures," last accessed April 20, 2010. [Online]. Available: http://www.digitalbond.com/index.php/research/scada-idsips/ids-signatures/

[12] A. Valdes and S. Cheung, "Communication pattern anomaly detection in process control systems," in *2009 IEEE International Conference on Technologies for Homeland Security*, Waltham, MA, May 11–12, 2009.

[13] P. Porras, M. Fong, and A. Valdes, "A mission-impact-based approach to INFOSEC alarm correlation," in *Proceedings of Recent Advances in Intrusion Detection*, October 2002, pp. 95–114. [Online]. Available: http://www.csl.sri.com/papers/mcorrelator/

[14] "Invensys process systems," last accessed March 23, 2010. [Online]. Available: http://www.ips.invensys.com/en/products/autocontrols/Pages/DistributedControl-IASeries-P018.aspx

[15] M. McDonald, J. Mulder, B. Richardson, R. Cassidy, A. Chavez, N. Pattengale, G. Pollock, J. Urrea, M. Schwartz, W. Atkins, and R. Halbgewachs, "Modeling and Simulation for Cyber-physical System Security Research, Development and Applications," Sandia National Laboratories, Tech. Rep. Sandia Report SAND2010-0568, Feb. 2010.

[16] "ICCP," last accessed April 21, 2010. [Online]. Available: http://intelligrid.ipower.com/IntelliGrid_Architecture/New_Technologies/Tech_IEC_60870-6_%28ICCP%29.htm

[17] "DATES demo at DistribuTech 2010," last accessed April 21, 2010. [Online]. Available: http://www.csl.sri.com/projects/dates/distributech.html

[18] J. Eisenhauer, P. Donnelly, M. Ellis, and M. O'Brien, "Roadmap to secure control systems in the energy sector." [Online]. Available: http://www.oe.energy.gov/Roadmap_to_Secure_Control_Systems_in_the_Energy_Sector.pdf

[19] J. Stamp, M. Berg, and M. Baca, "Reference Model for Control and Automation Systems in Electrical Power," Sandia National Laboratories, Tech. Rep., 2005.

[20] "Linking the oil and gas industry to improve cybersecurity," last accessed March 23, 2010. [Online]. Available: http://www.cyber.st.dhs.gov/logiic.html

[21] "Tofino," last accessed April 20, 2010. [Online]. Available: http://www.tofinosecurity.com/products/Tofino-Firewall-LSM

[22] "Industrial defender," last accessed April 20, 2010. [Online]. Available: http://www.industrialdefender.com/