A Distributed Real-Time Event Correlation Architecture for SCADA Security

Yi Deng, Sandeep Shukla

Chapter 6

A DISTRIBUTED REAL-TIME EVENT CORRELATION ARCHITECTURE FOR SCADA SECURITY

Yi Deng and Sandeep Shukla

Abstract Supervisory control and data acquisition (SCADA) systems require real-time threat monitoring and early warning systems to identify cyber attacks. Organizations typically employ intrusion detection systems to identify attack events and to provide situational awareness. However, as cyber attacks become more sophisticated, intrusion detection signatures of single events are no longer adequate. Indeed, effective intrusion detection solutions require the correlation of multiple events that are temporally and/or spatially separated. This paper proposes an innovative event correlation mechanism for cyber threat detection, which engages a semantic event hierarchy. Cyber attacks are specified via low-level events detected in the communications and computing infrastructure and correlated to identify attacks of a broader scope. The paper also describes a distributed architecture for real-time event capture, correlation and dissemination. The architecture employs a publish/subscribe mechanism, which decentralizes limited computing resources to distributed field agents in order to enhance real-time attack detection while limiting unnecessary communications overhead.

Keywords: SCADA systems, event correlation, temporal-spatial correlation

1. Introduction

Supervisory control and data acquisition (SCADA) systems are essential to the control and management of operations in the critical infrastructure (e.g., electrical power systems, water and wastewater treatment facilities, oil and gas pipelines, transportation assets and industrial process environments) [12]. A SCADA system uses sensors to monitor various physical quantities of the system under control and report them in real-time to a SCADA master (or control center). The execution of a state estimation algorithm, followed by

the application of control laws, generate control inputs that are sent to field devices to manipulate the control settings of actuators. In the case of an emergency (e.g., abnormal behavior is detected), the SCADA system must execute contingency responses to restore the system. Advanced communications techniques are widely adopted in SCADA systems to ensure the accurate and timely transmission of sensor data and control inputs [16].

In most SCADA systems, administrators are able to monitor and manipulate the data generated by field devices remotely – even from their homes. Often, field devices installed in remote areas are connected to an integrated network, which eliminates manual surveillance and maintenance of the devices. The use of networking technologies provides convenience for system operators, increased productivity for maintenance personnel and greater efficiency for critical infrastructure asset owners.

However, the adoption of advanced communications and computing technologies increases the susceptibility to cyber attacks. Consider the notorious Stuxnet malware that targeted a nuclear processing plant in Iran. Stuxnet was designed with four zero-day attacks and highly complicated intrusion functionality. It spread indiscriminately through flash drives and targeted Siemens industrial control equipment running a specific version of Microsoft Windows [6, 10]. Stuxnet significantly increased concerns about SCADA security. Meanwhile, several industry consortia (e.g., ISAC), standards agencies (e.g., NIST and NERC) and government agencies (e.g., DHS) have developed publications that outline regulations, best practices and guidelines for securing SCADA systems from cyber attacks.

In traditional information technology (IT) systems, intrusion detection systems (IDSs) are deployed to detect network-borne attacks. However, intrusion detection associated with SCADA networks must augment or revamp traditional IDSs because events often occur in the physical system. In traditional IT systems, signature-based and anomaly-based IDSs are typically used to detect intrusions and malicious behavior based on predefined attack patterns and deviations from normal behavior, respectively [4, 11, 13]. Although IDSs have seen much success in traditional IT systems, there are some inherent disadvantages when they are employed in SCADA systems. First, most IDSs are not specifically designed for SCADA systems so they are incapable of analyzing SCADA-specific communications protocols. Second, depending on the application scenario, the number of reported events generated by an IDS can vary considerably. For example, under normal operating conditions, IDS events are reported at regular, specified intervals; however, during potential malicious events, the reporting rate may increase significantly. In the case of a SCADA system, an overwhelming number of events could potentially paralyze the system. Third, legacy SCADA systems have limited computing and communications resources; as a result, IDSs are unable to satisfy the real-time constraints imposed on SCADA systems. Consequently, it is important to design and develop SCADA-specific IDSs that meet the requirements of the operating environment.

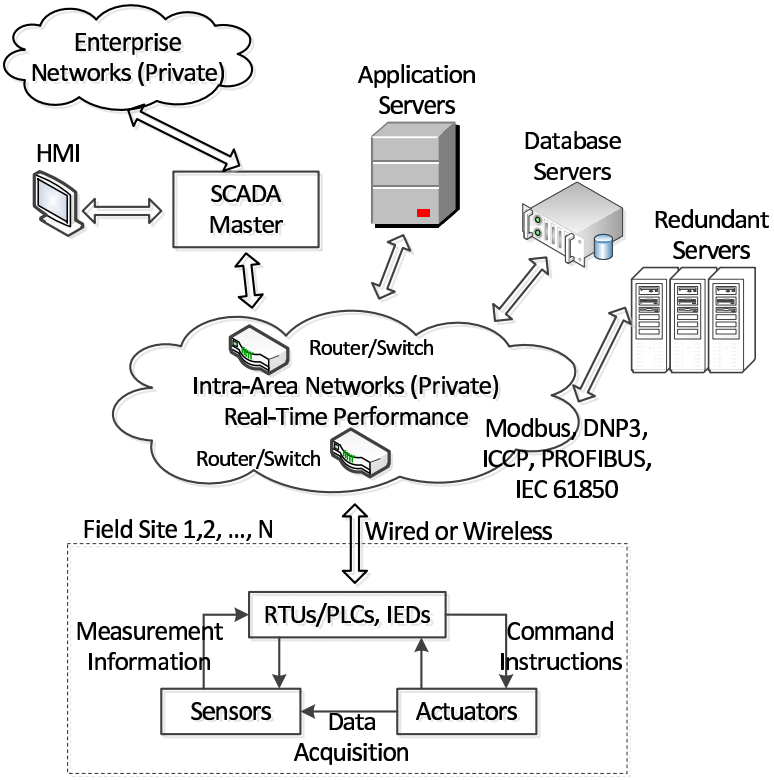

Figure 1. Typical architecture and components of a SCADA system.

2. SCADA System Architecture and Features

This section describes the architecture of a typical SCADA system and its principal features.

2.1 Distributed Architecture

A SCADA system is a mission-critical system that integrates advanced networking, computing and control technologies [19]. Although the scale of a SCADA system varies according to its application environment, it generally comprises three components: field site components, back-end platform components and a communications infrastructure (Figure 1).

The field site components generally include remote terminal units (RTUs), programmable logic controllers (PLCs), intelligent electronic devices (IEDs), remote sensors and actuators. RTUs are terminal devices that receive measurements from remote sensors and send instructions to actuators. PLCs are microprocessor-based controllers that can handle multiple inputs and outputs. IEDs are application-specific devices. In the electric power industry, IEDs mea-

sure voltages, currents, frequencies and phasor information, and are capable of tripping circuit breakers if anomalies or contingencies are detected.

The back-end components of a SCADA system typically include a human machine interface (HMI), SCADA master, application servers, application-specific databases and redundant servers. The HMI presents process data to human operators in a readable format. It can range from a single computer screen to a dedicated control center for supervising thousands of transmission lines at a major electric utility. The SCADA master provides a scalable, real-time computation framework. It gathers operational data from field devices and retransmits the data to application servers. The application servers implement decision-making applications that cannot be executed by field devices (e.g., wide-area state estimation, intelligent load shedding and intelligent islanding). The database servers archive the measurements and process data as well as provide data retrieval services for applications. Redundant servers provide backup services to other data servers to enhance system reliability.

The communications infrastructure consists of routers, switches and intra-area networks. Data is transmitted between the SCADA master and field site components at a pre-defined rate based on the real-time performance of the internal network. Various SCADA-specific communications protocols (e.g., Modbus, DNP3, ICCP, PROFIBUS and IEC 61850) are implemented in SCADA systems [8]. The networked infrastructure provides the benefits of cost, time and manpower savings, while enhancing situational awareness and control flexibility.

2.2 Unique Features of SCADA Systems

Unlike traditional enterprise IT systems, SCADA systems require enhanced reliability and availability, real-time performance, determinism, concurrency and security.

- **Reliability and Availability:** The requirements of system reliability and availability for process control systems are much higher than those for IT systems. Traditional IT systems can tolerate service outages (e.g., a webpage becoming slow or unresponsive) due to unstable hardware or an overwhelming number of access requests. In the case of control systems, system availability is critical. Furthermore, the designed mean time between failures (MTBF) for a SCADA system is much higher than that for IT systems. For example, the MTBF requirement for RTU modules is recommended to be greater than 720,000 hours [1, 5, 15].

- **Real-Time Performance:** SCADA systems must meet real-time requirements. When a contingency occurs, a SCADA system must detect and respond with the appropriate actions before events cascade to produce a large-scale physical impact. Compared with the real-time requirements of multimedia systems and virtual-reality systems [3], a SCADA system is required to meet strict, deterministic deadlines to ensure proper operation.

- **Determinism:** To safely manage a control system, SCADA systems require strict determinism to prevent random, uncertain or unknown states. A SCADA system requires that the monitoring system correctly reflect the underlying status of the control system. However, the increasing complexity and scale make the determinism difficult to achieve. Additionally, network communications present a challenge with regard to determinism. In the past, the network topology of a SCADA system was regular and relatively static [19]. An experienced system operator was familiar with the topology and the logical relationships between field devices. When a fault event occurred, the operator could infer the fault location by manually analyzing event types. Current SCADA network topologies, however, change dynamically based on the applications and operating requirements.

- **Concurrency:** The number of sensors varies according to the scale of a SCADA system. As such, SCADA systems must handle the concurrency that occurs in communications and computing. The concurrency of a SCADA system should be defined in a hierarchical manner. From a high-level perspective, the SCADA system must handle multithreaded and multitasked processes. From a low-level perspective, the field devices must handle multiple inputs and outputs while permitting scalability.

- **Security:** SCADA systems are commonly interconnected to public and enterprise networks. The interconnections subject SCADA systems to inherent threats associated with network communications. Additionally, the minimal processing power of field site components and limited bandwidth prevent the deployment of traditional security protection mechanisms such as anti-virus software. Moreover, SCADA systems have to operate correctly without interruption; the consequences of failure range from loss of revenue to personal injury or death.

Table 1 compares the properties of IT systems and SCADA systems [17].

3. Publish/Subscribe Paradigm

This section describes the publish/subscribe paradigm used for event correlation.

3.1 Event Correlation Techniques

Individual event analysis methods are typically adopted in order to enhance the security of SCADA systems. An event corresponds to an activity that is used to monitor, supervise or manipulate the system. Examples of events include a measurement uploading process, a remote login action into a field device, and a modification command issued by a system operator. Note that most operations in a SCADA system generate a series of events rather than a single event. For instance, a system operator may wish to check the operational

Table 1. Comparison between IT systems and SCADA systems [17].

Attributes	**SCADA Systems**	**IT Systems**
Availability	Extremely high	Low to medium
Integrity	Very high	Low to medium
Confidentiality	Low	High
Authentication	High	Medium
Time Criticality	Critical	Delays tolerated
System Life Cycle	15 years or more	3 to 5 years
Software Maintainability	Rare, informal, not always documented	Frequent, formal, documented
Interoperability	Critical	Not critical
Communications Protocols	DNP3, ICCP, Modbus, Fieldbus, PROFIBUS, BacNet	TCP/IP, UDP
Computing Performance	Very limited with older microprocessors	No limitation with new CPUs
Bandwidth	Limited	Very high
Administration	Centralized/localized	Centralized
Security Attack Impact	Process stability, equipment damage, environmental effects, personnel safety	Business impact

status of an IED located at a remote substation in order to check if the device needs maintenance. The request involves a series of events, such as logging into the master computer, invoking a maintenance application, establishing a communications channel, logging into the remote IED device, uploading history data, invoking the maintenance analysis application and recording the actions in the historical database. Based on these activities, event correlation can be performed to infer system behavior and identify the root causes of a problem or predict future trends.

Although sophisticated cyber attacks are usually stealthy and difficult to detect, they still induce anomalies and underlying events (e.g., abnormal packets, uncommon login attempts and sudden increase in traffic). If these events are analyzed individually, it is difficult to identify potential attacks. By utilizing event correlation techniques, however, an IDS can collect information from associated events to identify seemingly disparate attack actions. For example, when a SCADA system reports a login failure event, an event correlation engine would retrieve related events that occurred within a specified timeframe. If the number of login failures from the same IP address or the same region ex-

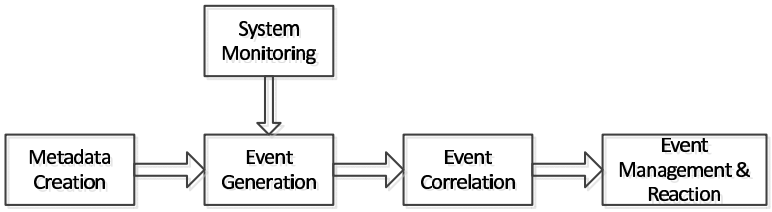

Figure 2. Processing flow in an event correlation system.

ceeds a predetermined threshold, then the event correlation engine can report a potential brute force login attack.

3.2 Event Correlation Flow

The general event correlation process involves a four-stage, pipelined data processing flow (Figure 2). In the first stage, Metadata Creation, distributed sensors create raw data. The remote sensors capture and measure analog signals and the status of systems in real time. Note that the sensors only generate unshaped metadata that is processed by field site components.

In the second stage, Event Generation, the field site components generate system events. In this stage, events are generated according to two situations. First, the field site components may generate events if abnormal behavior is detected in the metadata. The event generation process repackages the metadata according to a predefined event format and reports the events. Second, the field site components monitor the physical attributes of the system as it operates. Based on various requirements and configurations, the event generator identifies normal behavior event characteristics such as successful data transmission.

The third stage, Event Correlation, implements the event correlation engine in the SCADA master. Relying on application servers and database servers, the SCADA master collects all the events from the distributed field site components and uses correlation algorithms to analyze the events.

In the fourth stage, Event Management and Reaction, the event correlation engine sends results to higher-level event management applications. Since the events during this stage are dramatically decreased, only high-risk events are presented. The system operators responsible for interacting with applications perform the appropriate responses in the manual mode or in the automatic unattended mode.

3.3 Event Correlation Engine

Figure 3 shows the architecture of an event correlation engine. The engine has four main components: event queue, format decoder, event correlator, and rearrangement and output module.

When the dispersed events arrive at the event correlation engine, they are buffered to enable concurrent evaluation. Since SCADA systems have real-time

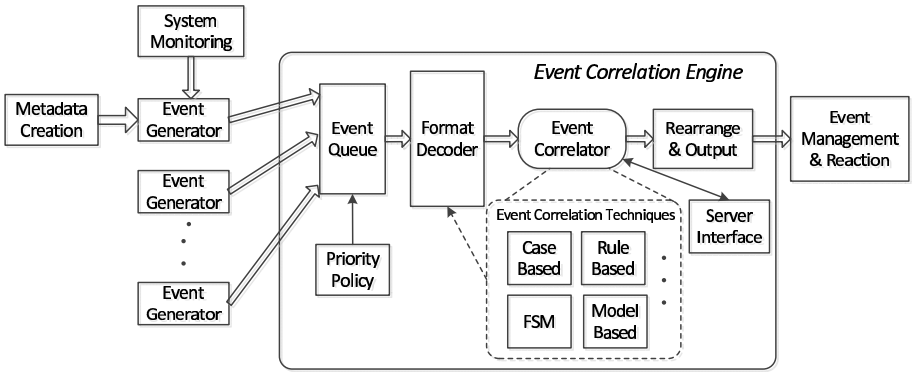

Figure 3. Event correlation engine architecture.

requirements, a priority management scheme is applied for high-risk events to ensure an adequate response time for emergencies. An event format decoder then extracts the effective segments from the primary event messages. After extraction, information is sent to the event correlator for analysis by event correlation algorithms. The event correlator interfaces with application servers and data servers to facilitate the retrieval of historical events. The event rearrangement and output module reformats the correlation results, packages statistical information and sends it to higher level applications.

3.4 Publish/Subscribe Mechanism

Event correlation techniques merge related events for evaluation. However, efficiency is a challenge for an event correlation engine. Commercially-available correlation engines (e.g., HP ECS, SMARTS and NerveCenter) typically have complex designs and user interfaces [9, 14, 18]. A primary reason is the requirement that a correlation engine must communicate continuously with every remote sensor. When remote field components upload primary events to an event correlation server, the event correlation engine buffers are filled with primary events, typically exhausting the resources.

We propose a publish/subscribe paradigm shown in Figure 4 to improve processing efficiency. The key aspect of the publish/subscribe paradigm is the integration of an authorized third party certification process in the event subscriber server. The event subscriber server alters the connections between the event correlation engine and the remote event generators from direct connections to subscription-oriented connections. This is a non-invasive online event correlation mechanism, which prevents interruptions of the remote field devices and event correlation server.

The publish/subscribe paradigm provides the possibility that the entire computing workload can be reduced by an order of magnitude. In this architecture, the distributed remote devices are considered to be event publishers and the event subscriber server is considered to be the subscriber. The event console

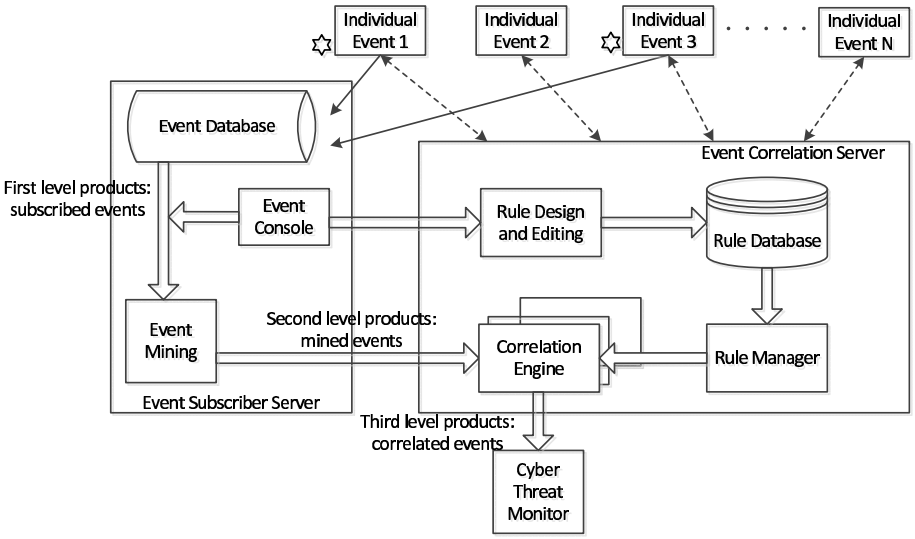

Figure 4. Automated publish/subscribe paradigm.

subscribes necessary information from different remote individuals and decisions can be made manually or automatically. Note that only the subscribed event publishers, marked with stars in Figure 4, are authorized to publish routine events to the event subscriber. This action decreases the amount of data transmitted, resulting in the more efficient use of the SCADA network. Additionally, partial workloads are allocated to remote devices, enhancing the computing performance of the event correlation server.

The event database is constructed and maintained as a two-dimensional data structure to simplify the event mining procedure. Patterns of the events that must be filtered are selected. Ultimately, a group of three-level hierarchical event correlation flows are constructed:

- **First Level Products:** Subscribed events.
- **Second Level Products:** Mined events.
- **Third Level Products:** Correlated events.

4. Temporal-Spatial SCADA Events Correlation

A two-dimensional (temporal and spatial) event correlator mechanism is incorporated to improve correlation accuracy. Note that temporal-spatial event correlation techniques have been used successfully in IT infrastructures [2, 7]. However, existing event correlation techniques are not well-suited to SCADA systems due to timing and availability requirements.

Figure 5 shows a two-dimensional temporal-spatial event correlator. In the temporal dimension, the event correlator tracks the unusual events within a

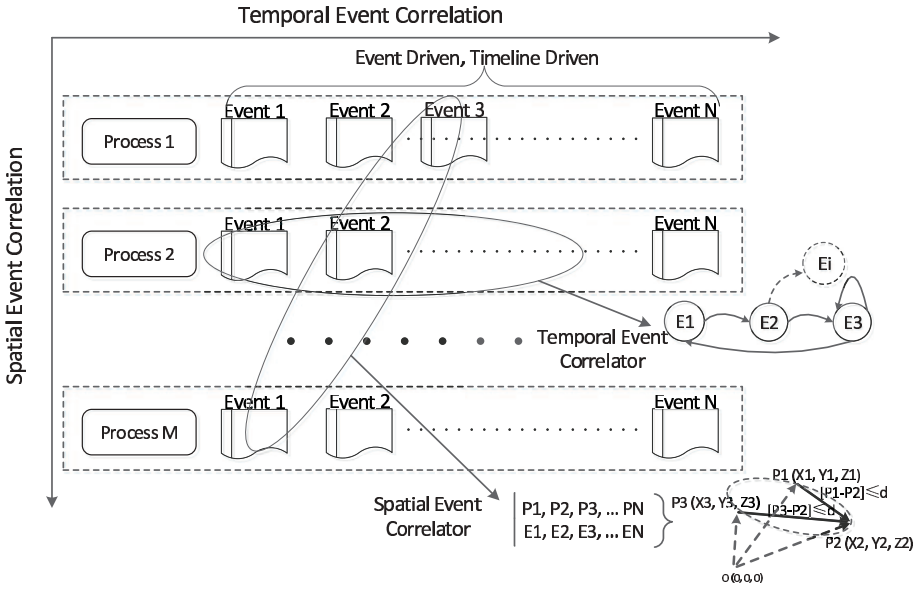

Figure 5. Two-dimensional (temporal and spatial) event correlator.

process using a sequential state machine $\{E1, E2, \ldots, EN\}$. As long as the state machine matches the predefined sequential rule patterns and reaches a final state Ei, the correlation engine triggers an event report to the correlator. For example, consider a scenario where the temporal event correlator is monitoring a repeat login attack where the attacker is attempting to access an account. The temporal event correlator enforces a limit on the number of failed attempts while abstracting the behavior as a state transition. When the sequential state machine reaches the final state, an alarm is triggered to indicate that the intrusion behavior has reached a high-risk level.

In the spatial dimension, the event correlator monitors the events that occur in the various processes and constructs multi-dimensional vectors $\{(P1, E1), (P2, E2), (P3, E3), \ldots, (Pi, Ej)\}$. Note that different processes can run on the same processor or different locations. Many attacks (e.g., distributed denial-of-service attacks) require coordinated actions. If the correlator were to detect the events individually, it would not trigger an event alarm. The proposed spatial event correlator, however, generates an index for each specific event and categorizes them into different security levels. The temporal-spatial event correlator automatically combines the two-dimensional process into a joint correlation procedure that prioritizes the tracking events.

5. Implementation

We designed and implemented the event correlation tools and integrated them in a server-based platform. The experimental SCADA system shown in Figure 6 was constructed to analyze the effectiveness of the publish/subscribe

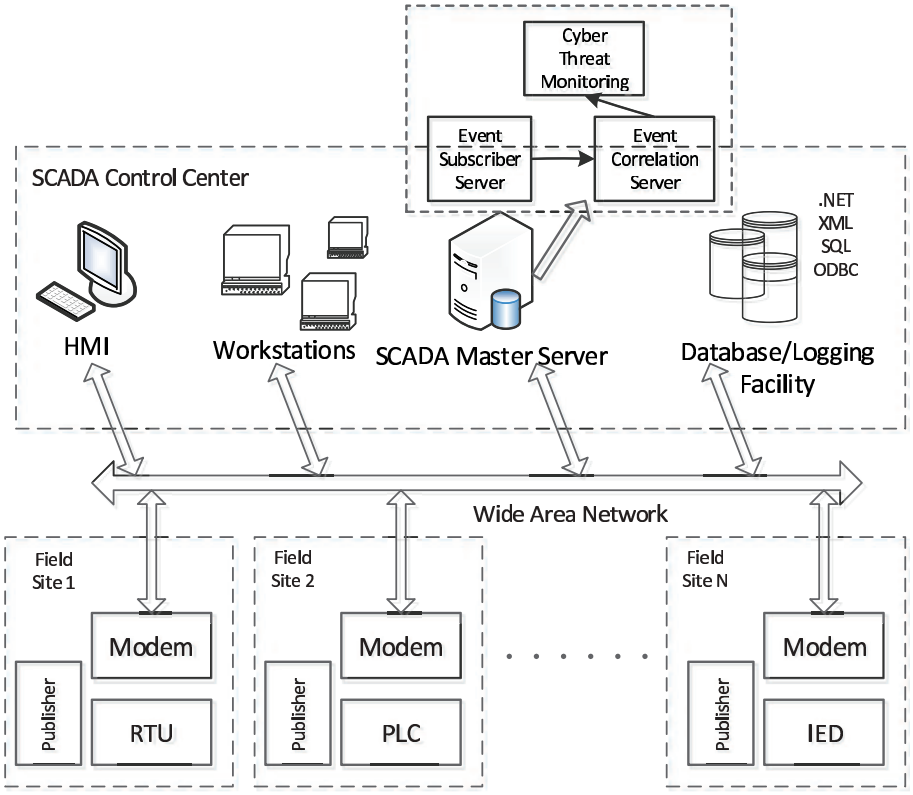

Figure 6. Experimental SCADA system.

temporal-spatial event correlation system. The experimental system deployed the publish function in every field device. The event subscriber server, the event correlation server and the cyber threat monitor module were implemented in the SCADA master server. Low-level communications between the SCADA master server and the distributed field sites were instantiated as TCP/IP. However, other SCADA-specific protocols such as DNP3 and IEC61850 could be implemented using software packages. The generality of the architecture enables the platform to be used as a dedicated SCADA testbed to assess network vulnerabilities and the effects of cyber attacks.

6. Conclusions

Event correlation is an effective mechanism for detecting cyber attacks. However, higher order event correlation requires detailed information about lower order monitoring and the event generation architecture. Data correlation is an essential method for evaluating every layer of attack detection, from the raw data layer to the reporting layer. The correlation quality can be improved by combining the temporal and spatial event properties within a joint correlation engine. Based on the design patterns and results, the publish/subscribe

paradigm used in conjunction with temporal-spatial event correlation appears to be an effective approach for detecting attacks on SCADA systems. Indeed, we hope that this work will stimulate renewed research focused on dedicated event correlation engines for SCADA systems.

References

[1] A. Bruce, Reliability analysis of electric utility SCADA systems, *IEEE Transactions on Power Systems*, vol. 13(3), pp. 844–849, 1998.

[2] J. Buford, X. Wu and V. Krishnaswamy, Spatial-temporal event correlation, *Proceedings of the IEEE International Conference on Communications*, 2009.

[3] G. Buttazzo, G. Lipari, L. Abeni and M. Caccamo, *Soft Real-Time Systems: Predictability vs. Efficiency*, Springer, New York, 2005.

[4] J. Cannady and J. Harrell, A comparative analysis of current intrusion detection technologies, *Proceedings of the Technology in Information Security Conference*, pp. 212–218, 1996.

[5] K. Erickson, E. Stanek, E. Cetinkaya, S. Dunn-Norman and A. Miller, Reliability of SCADA systems in offshore oil and gas platforms, in *Stability and Control of Dynamical Systems with Applications: A Tribute to Anthony N. Michael*, D. Liu and P. Antsaklis (Eds.), Birkhauser, Boston, Massachusetts, pp. 395–404, 2003.

[6] N. Falliere, L. O'Murchu and E. Chien, W32.Stuxnet Dossier, Symantec, Mountain View, California, 2011.

[7] G. Jiang and G. Cybenko, Temporal and spatial distributed event correlation for network security, *Proceedings of the American Control Conference*, vol. 2, pp. 996–1001, 2004.

[8] R. Kalapatapu, SCADA protocols and communication trends, presented at the *Instrumentation, Systems and Automation Society Conference*, 2004.

[9] LogMatrix, NerveCenter 6.0 Release Notes Windows and UNIX Version 6.0.02, NCRN60-02-03, Marlborough, Massachusetts, 2012.

[10] R. McMillan, Siemens: Stuxnet worm hit industrial systems, *Computerworld*, September 14, 2010.

[11] I. Nai Fovino, M. Masera, M. Guglielmi, A. Carcano and A. Trombetta, Distributed intrusion detection system for SCADA protocols, in *Critical Infrastructure Protection IV*, T. Moore and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 95–110, 2010.

[12] National Communications System, Supervisory Control and Data Acquisition (SCADA) Systems, Technical Bulletin 04-1, Arlington, Virginia, 2004.

[13] A. Patcha and J. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, *Computer Networks*, vol. 51(12), pp. 3448–3470, 2007.

[14] K. Sheers, HP OpenView event correlation services, *HP Journal Online*, article no. 4, 1996.

[15] Star Controls, Reliability and Availability of SCADA Systems, Shanghai, China (`www.star-controls.com/Files/ReliabilityandAvailabilityofSCADASystems.pdf`), 2010.

[16] K. Stouffer, J. Falco and K. Kent, Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security, NIST Special Publication 800-82, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[17] J. Weiss, *Protecting Industrial Control Systems from Electronic Threats*, Momentum Press, New York, 2010.

[18] D. Williams and D. Curtis, Magic Quadrant for IT Event Correlation and Analysis, Gartner RAS Core Research Note G00208774, Gartner, Stamford, Connecticut, 2010.

[19] B. Zhu and S. Sastry, SCADA-specific intrusion detection/prevention systems: A survey and taxonomy, *Proceedings of the First Workshop on Secure Control Systems*, 2010.