SAND2006-74227422, Printed November 2006

Applying New Network Security Technologies to SCADA Systems

Steven Hurd
Computer and Network Security Department
Sandia National Laboratories
P. O. Box 969
Livermore, California 94551-0969

Jason Stamp
Critical Infrastructure Systems Department

David Duggan and Adrian Chavez
Networked Systems Survivability and Assurance Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0672

Abstract

Supervisory Control and Data Acquisition (SCADA) systems for automation are very important for critical infrastructure and manufacturing operations. They have been implemented to work in a number of physical environments using a variety of hardware, software, networking protocols, and communications technologies, often before security issues became of paramount concern.

To offer solutions to security shortcomings in the short/medium term, this project was to identify technologies used to secure "traditional" IT networks and systems, and then assess their efficacy with respect to SCADA systems. These proposed solutions must be relatively simple to implement, reliable, and acceptable to SCADA owners and operators.

This page intentionally left blank.

Contents

Figures

List of Acronyms

AES	Advanced Encryption Standard
ANC	Adaptive Network Countermeasures
CCD	Center for Cyber Defenders
DOE	Department of Energy
DHS	Department of Homeland Security
GI	General Information
HIDS	Host-based Intrusion Detection System
HMI	Human-Machine Interface
IDART	Information Design Assurance Red Team
IDS	Intrusion Detection System
IED	Intelligent Electronic Device
I/O	Input/Output
IP	Internet Protocol
IPS	Intrusion Protection System
IPsec	Internet Protocol Security
ISA	Instrumentation, Systems and Automation Society
IT	Information Technology
LAN	Local Area Network
NIDS	Network-based Intrusion Detection System
NIST	National Institute of Standards and Technology
NSTB	National SCADA Test Bed
OAID	Open Architecture for Interoperable Design
OE	Office of Energy Delivery and Reliability
PCS	Process Control System
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SLAP	Secure Linux Appliance for Process Control Systems
SSH	Secure Shell
TSWG	Technical Security Working Group
VPN	Virtual Private Network

Introduction

Supervisory Control and Data Acquisition (SCADA) systems for automation are very important for critical infrastructure and manufacturing operations. SCADA systems collect and transmit information between sensors, controllers, and central management stations; concurrently they store, process, and analyze information. They have been implemented to work in a number of physical environments using a variety of hardware, software, networking protocols, and communications technologies.

One common thread among SCADA systems is that they were developed without adequate regard for security issues. Definitely, many legacy SCADA systems were implemented in an era before security issues became a major concern. However, even for contemporary SCADA technology, the most important factors appear to be (1) the lack of perceived business case for SCADA security features, and (2) the reluctance of SCADA operators to implement security features that might impede the operation of the SCADA system. Furthermore, as SCADA systems become increasingly reliant upon commodity PC hardware, operating systems and software as well as network devices and protocols, it is clear that SCADA devices will be subject to an ever increasing amount and variety of threat.

Research at Sandia and elsewhere has focused on how to improve security and reliability for next-generation SCADA systems over the long-term. However, as SCADA systems are often attractive targets for adversaries and replacement cycles generally range in decades, rather than months or years, there clearly was a critical need to identify ways to address security shortcomings in the short- and medium-term.

This project investigated how technologies developed to secure conventional information technology (IT) networks could be applied to address securing SCADA systems and networks. We accomplished this by bringing together:

- Some of the world's best operational network defense and intrusion detection analysts,
- Intrusion detection, adaptive network countermeasures, and encryption researchers,
- SCADA security assessment experts.

This project's end goal was to identify and test technologies to significantly improve the security posture of existing SCADA systems. These proposed solutions must be relatively simple to implement, reliable, and acceptable to SCADA owners and operators.

The major goals (each with multiple milestones) for the project were as follows:

- Characterize SCADA systems and their associated vulnerabilities,
- Examine security solutions from conventional IT networks and systems,
- Merge SCADA vulnerability information with identified solutions,
- Prototype and test the identified solutions.

This page intentionally left blank.

Goal: Characterize SCADA systems and their associated vulnerabilities

This goal had three milestones associated with it. Each of the milestones were intended to validate our understanding of the uses of SCADA systems and the vulnerabilities that currently exist, due to the way the systems are implemented. These milestones were:

- Identify a representative sample of SCADA systems (equipment and operations),
- Verify the representative sample of SCADA systems with industry,
- Identify and categorize SCADA systems vulnerabilities.

Each of the milestones is discussed in the following subsections of this paper.

Milestone: Identify a representative sample of SCADA systems (equipment and operations)

This milestone was accomplished early in the work for this LDRD and we strongly leveraged the paper entitled “A Reference Model for Control and Automation Systems in Electric Power” [1], which is an excerpt from a larger paper entitled "Automation Systems Reference Model" that was developed by Sandia using National SCADA Test Bed (NSTB) funding. (The latter paper remains unpublished by NSTB.) A diagram from this paper, "Object-Role Model for Electric Power and Automation Systems," has been included in this report (Figure 1, following).

At their essence, the majority of SCADA systems utilize the following types of equipment and services:

- Infrastructure Equipment (this includes actuators, sensors, and field Input/Output (I/O) devices, as well as the actual physical assets being managed),

- SCADA Field Equipment (this includes I/O Controllers, such as Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IEDs), and local automated control capabilities),

- Control Center (this includes system status data, system-wide automated control, system management functionality, historical data, alert systems, and exported data),

- Automation Oversight (this includes connections to business systems, oversight entities, or partners).

For the purposes of our research, we hypothesized that the communication connections between SCADA Field Equipment and Control Centers were the aspect of a representative SCADA system that would most benefit from the application of IT security principles, as each of the other communications connections (between control centers and business, oversight, other control centers, and the like) has been addressed by other work [2].

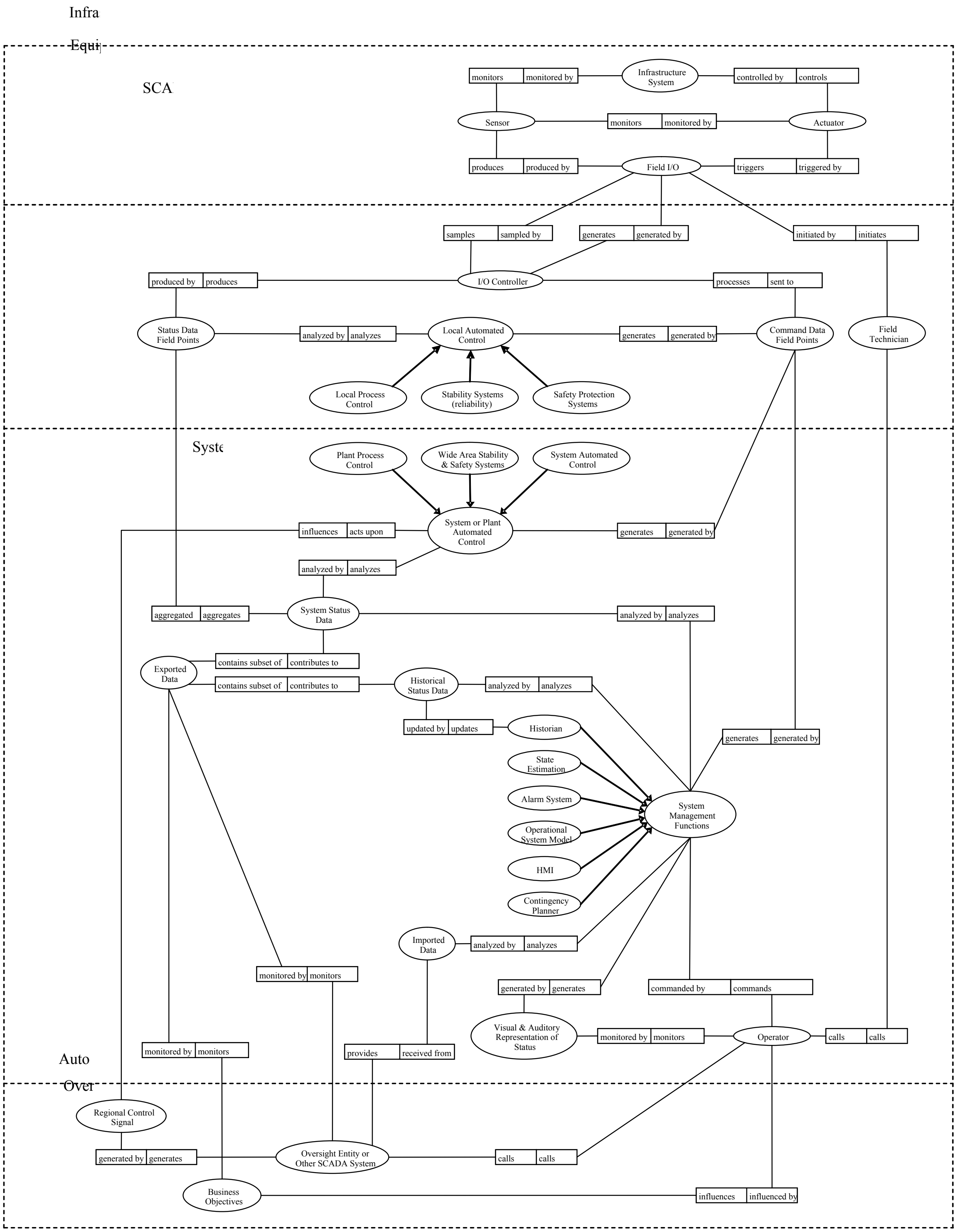


Figure 1: Object-role model for electric power control and automation systems. [1]

Milestone: Verify the representative sample of SCADA systems with industry

This milestone was accomplished through several conversations with SCADA system owner/operators during early 2005 and through presentations made at SCADA conferences. Most prominent was the Clemson Power Systems conference in March 2005, where Sandia involvement in the organization of the conference allowed the opportunity for a three-hour session on the topic. Subsequent discussions with industry members from both the oil & gas and electric sectors reinforced the accuracy of our target SCADA architecture.

Our “representative sample” of SCADA was verified as being accurate. Furthermore, our hypothesis (that the network connections between SCADA Field Equipment and the Control Center were the aspect of a representative SCADA system that would most benefit from the application of IT security principles) was also verified as accurate by the same sources.

Milestone: Identify and categorize SCADA systems vulnerabilities

This project leveraged work being done in another project funded by the Technical Security Working Group (TSWG). The TSWG project developed a guidebook for security in automation systems that formed the basis of an effort on this milestone to identify and categorize SCADA system vulnerabilities. Additional work was necessary to distill information from the guidebook into a more compact form. Sandia provided much of the technical input to the guidebook. [3]

The US-CERT web site, "Overview of Cyber Vulnerabilities" [4] includes a list of standard vulnerabilities in control systems, including:

- Access to the Control System Local Area Network (LAN),
- Common Network Architectures,
- Dial-up Access to the RTUs,
- Vendor Support,
- IT Controlled Communication Gear,
- Corporate Virtual Private Networks (VPNs),
- Database Links,
- Poorly Configured Firewalls,
- Peer Utility Links,
- Sending Commands Directly to the Data Acquisition Equipment,
- Exporting the Human-Machine Interface (HMI) Display,
- Changing the Database,
- Man-in-the-Middle Attacks.

This work largely echoed the previous work from the document entitled "Common Vulnerabilities in Critical Infrastructure Control Systems" [5]. Of this list, a security solution that protects against each of these (excepting the "Peer Utility links" - already established as out-of-scope for this effort) was desired.

This page intentionally left blank.

Goal: Examine security solutions from conventional IT networks and systems

The intent of this goal was to identify the current status of the industry in practice and to research efforts in securing SCADA systems. To accomplish this goal, there were two milestones defined. The milestones below break down those efforts.

Milestone: Identify best practices for installation, configuration, and operation.

An initial model was developed that enumerates different methods used to secure conventional IT networks and systems. These methods encompass technologies, configuration, or processes used to secure networks and systems. From that point, critical attributes were documented for each protection mechanism (such as articulating the addressed threats, efficacy, reliability, and new problems resulting from the use of the protection mechanism).

The conventional IT network and system security model revolves around providing assurance in three primary areas:

- Confidentiality: Confidentiality is assurance that information is not disclosed to unauthorized persons, processes, or devices [6].
- Integrity: Integrity is the protection of information against unauthorized modification or destruction [7].
- Availability: Availability is the timely, reliable access to data and information services for authorized users [6], or the assurance that authorized users can access the information necessary to complete their jobs [7].

Thus, best practices from conventional IT networks and systems address some or all of these three areas.

- Encryption/VPNs (Confidentiality): Encryption is a method whereby information is encoded in such a way that only intended parties should be able to decode it. While the focus with respect to encryption has traditionally been to protect communication links (data in motion), due to recent losses of personally identifiable information from laptop computers, many organizations have rushed to deploy disk/file encryption solutions (data at rest).

  VPNs are used to link a host (or a network) to a remote network. Network traffic is encrypted between the host (or network) and the remote network. Once traffic arrives at the remote network, it is decrypted and forwarded to the appropriate system.

  Encryption algorithms are usually divided into two types: Symmetric-key and Asymmetric-key. Symmetric-key algorithms are those who use the same key for encryption and decryption (e.g. “a shared secret”), or by knowing the encryption key, you can calculate the decryption (and vice versa). Asymmetric-key algorithms often referred to as Public/Private Key algorithms, have been established such that knowing the public key will provide no information as to the content of the private key. For confidentiality purposes, a user will typically use some other user's public key to encrypt information. That way only that other user can decrypt the information (using their private key).

  - Concerns include the loss or damage of encryption/decryption keys (and resultant data loss) and denial of service (if an encrypted connection can't be properly established or maintained).

- Integrity Checking (Integrity): The most common way to perform integrity checking is to apply a cryptographic function called a "hash function" to represent the file (or information) to a relatively unique, fixed length value, called a "hash". If the current hash value differs from a previously computed hash value, it is reasonable to assume that the file (or information) has changed. Certain host-based intrusion detection programs, such as Tripwire, will point out changes in certain sensitive files as a mean to identify intrusions.

  - Concerns include interruptions caused by false positive alerts, as these alerts could distract operator attention from other pressing issues.

- Network Perimeter Defenses (Confidentiality, Integrity, and Availability): The most commonly-used device in this category is the firewall. Firewalls are network devices that will block (or allow) network traffic (subject to the rules applied by a system administrator). Early firewalls could block (or allow) traffic based upon Internet Protocol (IP) address or source/destination port (port numbers were typically associated with certain application-level protocols). Newer firewalls are aware of the current state of a network connection and can block traffic that is not consistent with that state.

  Application proxies are another commonly deployed network perimeter defense. Proxy servers act as an intermediary between a client and server. They are aware of the application protocol and are typically configured to block traffic that falls outside the protocol's specifications.

  - Concerns include the fact that configuring and verifying the proper configuration of firewalls and application proxies is not easy. The biggest concern is that legitimate traffic will be blocked due to firewall or proxy misconfiguration. While it is a concern when unwanted traffic is forwarded by the firewall due to firewall misconfiguration, this problem certainly exists if no firewall is in place.

- Network Intrusion Detection and Prevention Systems: Network-based Intrusion Detection Systems (NIDS) monitor traffic within a network. The most commonly used NIDS are configured to raise an alert based upon traffic that matches a pattern or signature. Other NIDS use different methods (such as heuristics) to identify anomalous traffic. NIDS are responsible for alerting, but not reacting to alerts. Thus, Intrusion Prevention Systems (IPS) were introduced. IPS are designed to automatically react to NIDS alerts by cutting off access between attackers and target systems.

  - The largest concern with NIDS alerts is dealing with false positives and having staff with sufficient expertise to differentiate between anomalous, non-malicious traffic and truly malicious traffic. In IPS, the prime concern is that it is possible for legitimate traffic to be blocked by accident as well as for an attacker to use the IPS as a denial of service attack.

- Anti-replay measures (Integrity): These measures are designed to prevent an attacker from recording valid information (typically protected by encryption) and acting as the original sending and replaying this information later. These measures typically involve including time stamp information or a counter into what is encrypted. With anti-replay measures in place, the receiver will know whether a message is "original" or "replayed".

  - The primary concern is denial of service at such time the counter or time stamp becomes out of sync.

- Authentication (Confidentiality and Integrity): Simply put, an authentication mechanism is designed to allow a user (or machine) to "prove" they are who they claim to be. There are many different methods of authentication, ranging from a user being issued a password challenge (for a shared secret) to two systems mutually authenticating through the use of digital certificates. At this time, biometrics and multi-factor, token-based authentication is beginning to replace traditional passwords.

  - The primary concern is denial of service if a user forgets a password or a digital certificate is lost.

- Access Controls (Confidentiality and Integrity): Typically used in conjunction with authentication measures, access controls are ways to control which users or systems can read, write, and/or execute certain files, or take certain actions on a system.

  - The primary concern is that misconfigured access controls will deny service to legitimate users.

- System Management Practices (Confidentiality, Integrity and Availability): System management practices are designed to make systems as reliable and impervious to attack as possible. The most important system management practices are: Configuration management, patching/update strategies, backup processes, and the use of anti-virus/anti-malware software.

  Configuration management is an essential component of a well-managed system.

Keeping current and historical configuration information (such as applications installed, versions/patch levels and open services) is vital.

Even the best software still has flaws that are uncovered after its release. These flaws can range from certain functions not working as advertised to avenues an attacker can exploit to bypass security measures. In addition to fixing flaws, software developers often add helpful features in upgrade releases. Accordingly, software patches and upgrades play a vital role in preserving system integrity and increasing functionality. However, it is quite possible that applying a patch will affect a system's operating status. Furthermore, a newly applied patch may be incompatible with other software on a system. Thus, it is important to have patch management practices that identify incompatibilities before application as well as schedule patch application appropriately (often based upon the severity of the flaw being patched).

System crashes and the accidental or intentional deletion of critical files are facts of life. Accordingly, effective backup processes are important. These processes include: the nature of the backup (incremental, full, etc.), frequency of backup, length of time backups are kept, and periodic verification of backups (through restoration of files).

Computer viruses and other instances of malware are ever-present threats to computers. Commercial anti-virus/anti-malware software is very effective at minimizing the exposure to these threats.

  - Concerns in this area include: Configuration management's primary downsides are the effort required to gather and maintain accurate information and the risk associated with acting on stored configuration management information that turns out to be incorrect. Applying patches or upgrades to a stable system has the possibility to introduce instability, incompatibilities, or new system vulnerabilities. In addition, if not scheduled appropriately, the patching or upgrading process can result in a denial of service to legitimate users. The biggest problems with backup software are breaches of confidentiality (if tapes are stolen). Anti-virus/anti-malware software is notorious for taking system resources at inopportune times. Please note: Recommendations regarding the use of anti-virus software in control systems can be found in the joint Sandia/National Institute of Standards and Technology (NIST) report, "Using Host-based Anti-virus Software on Industrial Control Systems: Integration Guidance and a Test Methodology for Assessing Performance Impacts "[8].

- Logging (Integrity): It can also be used to identify instances where confidentiality has been breached. The idea of logging is simply to generate an audit or activity trail that can be reviewed regularly. Log information can not only provide information about a system breach but also indications that a system is misconfigured or operating erratically.

    - The primary concern is that depending on the severity of a system compromise, local logs are of little or no use, as they could be erased by an attacker. Worse yet would be logs that pointed forensic investigators in the wrong direction.

- Defense in Depth/Breadth (Confidentiality, Integrity and Availability): The most successful approach to security has been when multiple complimentary security approaches are used. Defense in depth typically refers to multiple security measures in series (such as physical access controls in conjunction with authentication), while defense in breadth typically refers to covering atypical system access methods.

    - The primary concerns involve denial of service and system complexity. If multiple security barriers are placed in series, any barrier that accidentally blocks access is problematic. Thus, multiple barriers would seemingly present more opportunities for overall system denial of service. System complexity is seen as making problem resolution more difficult.

Milestone: Examine research efforts.

This effort is to determine which security benefits would occur by leveraging research efforts for SCADA or IT security (in contrast to the project's main goal of applications of established IT technology).

Adaptive Network Countermeasures (ANC) is a research effort at Sandia, California designed to thwart or greatly complicate outside reconnaissance efforts (network and system enumeration), through the use of intelligent, adaptive countermeasures [9].

The ANC model incorporates Intrusion Detection Systems (IDS) and General Information (GI) modules that provide information regarding IDS alerts as well as system and network configuration to a decision-making module called “Athena”. In turn, Athena controls actions taken by response modules that include firewall change processes and deception systems (that utilize honeyd).

However, although ANC is an attractive option for traditional IT networks, the complexity and relative immaturity of ANC and the fact that it is focused strictly on enumeration, coupled with SCADA owner/operator reliability requirements, make ANC a poor choice for use with SCADA systems at this time. Basic Intrusion Prevention Systems (IPS) would likely provide the majority of ANC’s value with a small fraction of the potential problems.

This page intentionally left blank.

Goal: Merge SCADA vulnerability information with identified solutions

The three milestones that apply to this goal have the intent of using internal and external expertise to develop a candidate security architecture that will mitigate the current vulnerabilities and security concerns found in today's SCADA systems.

Milestone: Assemble an expert team to integrate security into SCADA

A team was formed whose members had expertise in SCADA operations, SCADA security, and conventional IT security. This team provided the information necessary to write the paper entitled "Fundamental Security Practices for Control and Automation Systems in Electric Power" [10] (while already available as a SAND report, it is being considered for publication in an Instrumentation, Systems, and Automation Society (ISA) book on Process Control System (PCS) security). While this paper did not cover all the best practices uncovered through the work in the previous Goal, it contained a significant subset of those best practices.

Milestone: Initial security architecture completed

Using SCADA Test Bed facilities at Sandia, the team developed requirements for the necessary security architecture for SCADA systems. These requirements included:

1. Encryption & data authentication: For secure operational communications, authenticated communications between field devices and plants/control centers is critical.
2. Logging & forensics support: A key weakness of conventional SCADA systems is their inability to support adequate logging for security monitoring or forensics. Any viable solution that we propose must address this.
3. Intrusion detection & prevention: Intrusion Detection Systems (IDS) are a key element of security monitoring; adapting IT technology to SCADA will ideally include distributed sensors for IDS.
4. Firewall and network filtering: Given their narrow range of acceptable use, SCADA systems represent an ideal opportunity to deploy permit-by-exception network filtering to enhance security.
5. Encrypted serial communications: To secure legacy systems, an upgraded, packet-based network between field sites and plants/control centers is presumed. (This assumption leaves out truly legacy systems, often with 2400 baud or lower speed communication capabilities, in favor of a technology space with adequate bandwidth overhead to allow for IT-style technology insertion.) Adding serial encapsulation capability to the security architecture allows for its introduction into older systems.
6. Authentication and logging for remote access: Remote access for configuration of PCS devices is easily accomplished using little or no authentication. The new

secure architecture will ameliorate this deficiency by adding stronger authentication mechanisms, which will be assiduously logged.

7. Control system visualization & monitoring: Finally, the candidate PCS security architecture includes multiple steams of security monitoring data which should be archived, and a database will be developed to support this requirement. Furthermore, the operational nature of the control center environment (including human monitoring) is well-conditioned for the deployment of an on-line security visualization tool to support real-time monitoring.

Milestone: Verify solution viability with industry

This milestone is expected to be gathering its information from several sources. First, it is expected that as the “Best Practices” paper becomes more widely read and put into practice, there will be comments provided by those that are using it. Those comments could affect the architecture. The other verification step is by our presenting the candidate security architecture to several industry groups for feedback. This was done in September 2005 at the KEMA Security Conference held in Albuquerque and also at the 2006 Clemson Power Systems Conference in Clemson, South Carolina. The general elements of the proposed security architecture were approved, and particular comments about specific technological issues were incorporated as modifications to the architecture.

Goal: Prototype and test the identified solutions

This last goal was made up of four milestones that make up the bulk of the project. Building an actual prototype and performing all the tests and evaluations that goes with it was done in this part of the project.

Milestone: Build secure SCADA prototype

Using previously published materials from this project, a prototype solution was developed to solve a portion of the security problem for SCADA environments. This prototype contains two components, one that participates in the actual security of the system and the other that will help determine the health and status of the security components of the system. The first component is a general purpose security appliance, in the form of an embedded Linux computer that provides security services to the overall system. The second component utilizes existing visualization software that was modified to work with SCADA-specific network protocols. These two components are described further in a later section of this paper.

Milestone: Testing and iterative refinement

As with development of any complex prototype, this prototype has had to be refined in its operation as it was tested in an operational environment. That is to be expected and was planned for in this project. Several iterations were necessary to ensure that the prototype actually solves more problems than it creates with regard to security and operational functionality in the SCADA environment. It has been, and will continue to be, tested within the National SCADA Test Bed facility. Utilizing test procedures, we are able to identify areas needing improvement and feed that information back to the design team for changes.

Milestone: Information Design Assurance Red Team (IDART) testing

Informal red teaming of the security components within the prototype has occurred throughout the development process. Beginning in July 2006, a red team comprised of student interns from the Center for Cyber Defenders (CCD) at Sandia, California was tasked with performing a series of security tests with the SLAP device. While some deficiencies were initially found, they were limited to misconfigurations resulting from the current SLAP configuration process. As soon as all misconfigurations were fixed, the SLAP device was found to perform in a secure and efficacious manner.

As the SLAP device continues to evolve through on-going development, it will be critical to continue to perform security tests.

Milestone: Compile and report results

This milestone is the final one for this project, and was the one that has lasted the longest. Information for this report has been gathered all through the process in order to ensure that the project was sufficiently documented.

This page intentionally left blank.

Secure Linux Appliance for PCS (SLAP)

This section describes the development and implementation of the Secure Linux Appliance for PCS (SLAP) Device as a principal component of the security architecture.

Providing effective security services to legacy SCADA devices that does not degrade performance is a key goal in the Department of Energy (DOE) and Department of Homeland Security's (DHS) "Roadmap to Secure Control Systems in the Energy Sector" [11]. In this report, they write,

> "Communication between remote devices and control centers and between business systems and control systems is a common security concern that requires secure links, device-to-device authentication, and effective protocols."

The value of creating a "bump in the wire" security appliance was evident to the co-Principal Investigators of this research project in early 2004. This belief was confirmed and strengthened through a series of conversations with SCADA systems owner/operators.

As this device was primarily responsible for securing the communication path between SCADA devices or a local network of SCADA devices, it was evident that a pair of these appliances (one at each end of the connection) would be necessary. However, it was important that a single appliance in a control center could communicate with a number of different appliances deployed at locations in the field.

The challenge was simple: Design and build an extremely reliable device that would provide SCADA systems with the most important security features utilized by conventional IT systems and networks. These include:

- Device-to-Device Network and Serial Encryption
- Network-based Intrusion Detection
- Host-based Intrusion Detection (for the appliance itself)
- Firewall Capabilities
- Device-to-Device Authentication
- User Authentication Services
- Logging Capabilities
- Remote Management Capabilities

This device would need to be capable of being "dropped into" existing SCADA implementations, such that existing SCADA systems would not experience any operational impact or require changes (except for negligible latency introduced by inserting the SLAP into the system).

In addition, as opposed to many vendor solutions that focused on proprietary solutions, the SLAP would be based entirely on open-source software and standardized hardware, using an open architecture to promote interoperability.

SLAP Design

The SLAP was initially implemented on a small footprint embedded computer running Debian Linux, pictured in Figure 2.

- PC104 architectural & industrial enclosure
- Hardware-accelerated encryption (on the Ethernet interfaces)
- 533MHz XScale processor
- 2 Ethernet & 4 serial connections (expandable)
- 8 binary status inputs (expandable)
- 16MB or 32MB Flash ROM
- CompactFlash slot for additional storage

Figure 2: Original SLAP System

The operating system and software used on this device included:

- Embedded Linux (Debian)
- Internet Protocol Security (IPsec), that uses Advanced Encryption Standard (AES) Encryption) for network encryption
- SCADAsafe for serial encryption
- “Snort” for network intrusion detection
- IPtables for firewall capabilities
- Syslog-ng for logging capabilities (both local and remote)
- Secure Shell (SSH) for remote management access to the appliance
- A simple host-based intrusion detection system (HIDS), that utilizes “find”, “sha1sum/md5sum” and “logger”).

Development on the device began in summer 2005. Testing of the initial proof-of-concept began in fall 2005. After iterative refinement, public demonstrations of capability began in spring 2006.

Although the initial solution was effective, there were a sufficient number of annoyances working with this hardware to warrant investigating other hardware platforms. Thus, the team began working with a second, more capable hardware platform, shown in Figure 3.

- Mini-ITX board and fanless enclosure
- 1GHz VIA processor
- 2 Ethernet & 6 serial connections (expandable)
- PCI expandability
- 512MB Flash ROM
- 1GB RAM

- Virtually the same OS & software from the first hardware platform

Figure 3: An assortment of Mini-ITX based SLAP devices

The team found this platform to have far fewer annoyances and better overall performance and capabilities.

In recent months, team members have implemented a proof-of-concept of the SLAP architecture using Gentoo Linux (rather than Debian Linux) on a Shuttle PC platform. The belief was that porting the SLAP architecture to a different Linux distribution would be a relatively quick and easy process. This belief was determined to be valid, as the porting process took less than one week to complete.

This page intentionally left blank.

SCADA Visualization

Certainly it is important for a SLAP device to function properly when working in tandem with another SLAP device. However, if SLAP devices are to be widely deployed within a large, geographically distributed SCADA system, it is critical to be able to aggregate information from among the SLAP systems & present a consolidated view to operators. Thus, as a part of our effort to build a prototype SLAP device, we designed a method for the log information from each SLAP to be aggregated into a database along with an application to monitor the database for critical events and present them graphically to operators.

When implementing logging capabilities on the SLAP device, we chose to use the syslog-ng service. This service, which is an enhanced version of the Unix standard "syslog" service, is capable of transmitting log information to a centralized syslog-ng server. We use syslog-ng in conjunction with a MySQL database to capture the raw log information of all events received (from all SLAP devices) and place them in a single database table. Raw entries are then exported into a formatted database that better supports real-time interaction. This process is graphically depicted in Figure 4.

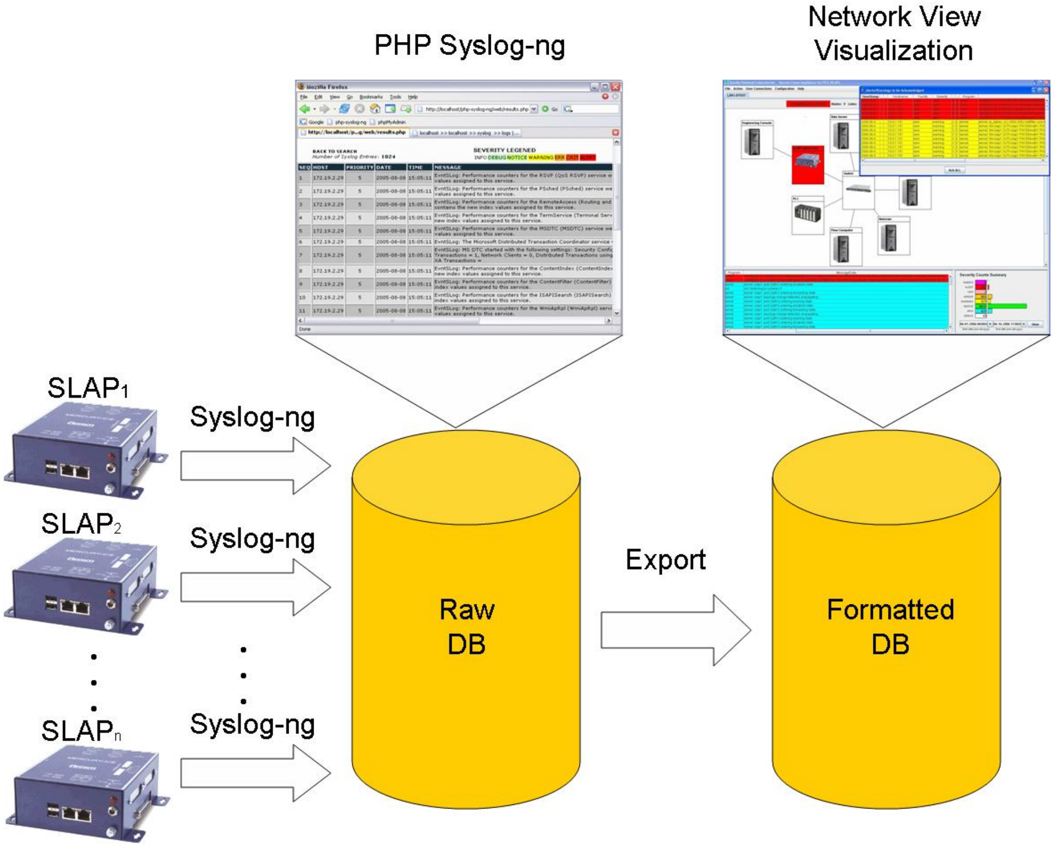
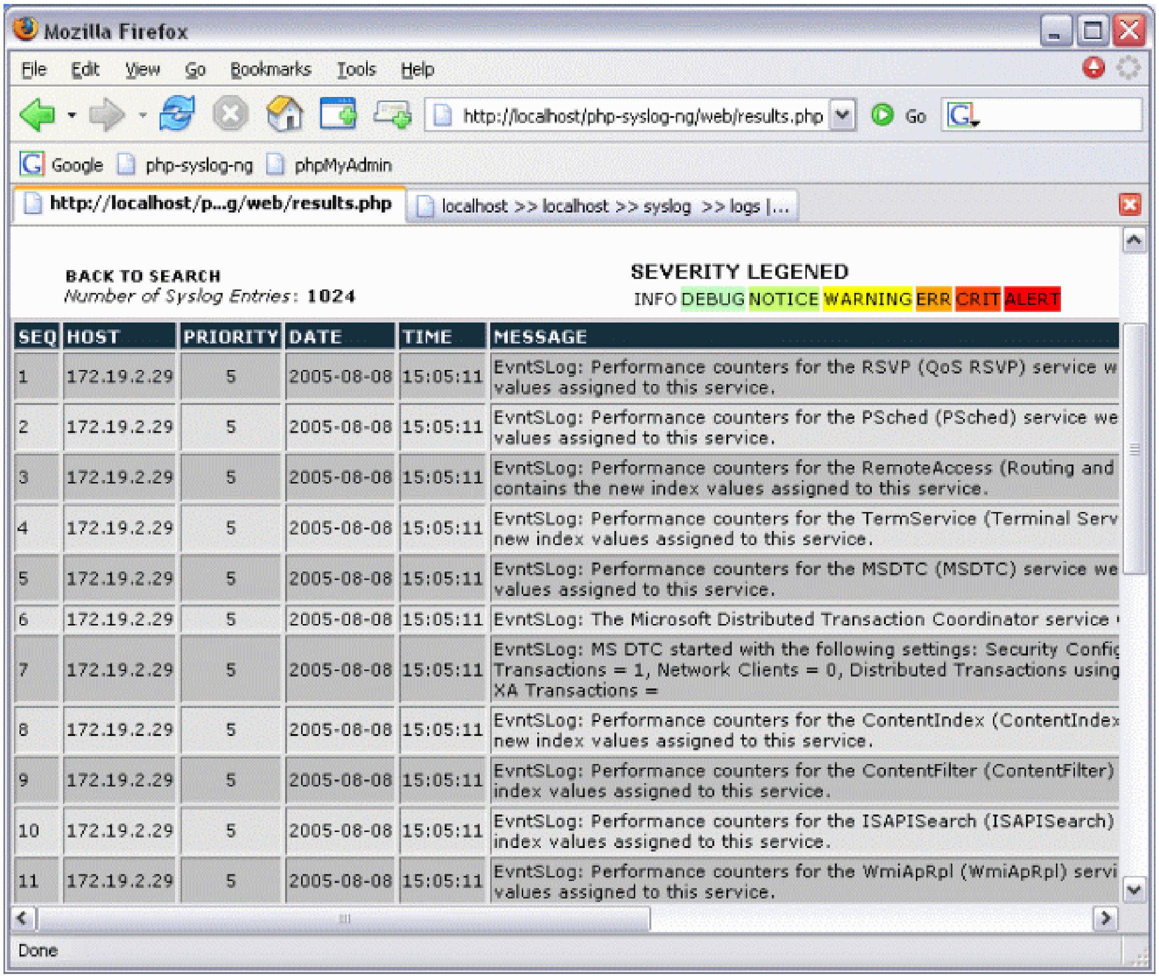


Figure 4: The data aggregation and visualization process

In addition to the data collection and storage, this centralized system utilizes a web server to present pertinent information to operators. The web server hosts PHP-based web pages that allow the information to be translated in tabular form, with color-coding to indicate the severity of the event. These pages support filtering and searching on specified criteria, such as severity level, service, date/time, etc. An example of this tabular view of log entries can be seen in Figure 5.


Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Go
Google
php-syslog-ng
phpMyAdmin
http://localhost/p...g/web/results.php
localhost >> localhost >> syslog >> logs |...
Done

http://localhost/php-syslog-ng/web/results.php

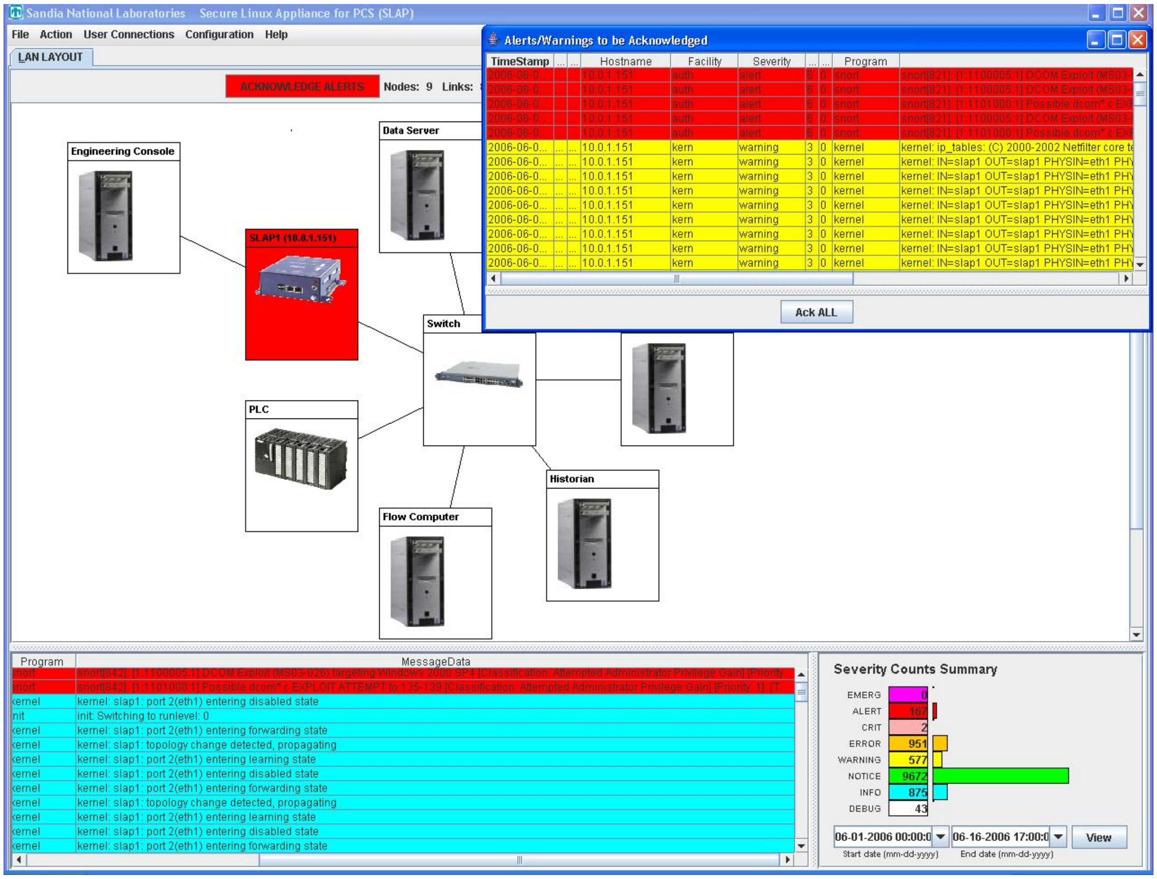
BACK TO SEARCH
Number of Syslog Entries: **1024**

SEVERITY LEGENED
INFO DEBUG NOTICE WARNING ERR CRIT ALERT

SEQ	HOST	PRIORITY	DATE	TIME	MESSAGE
1	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the RSVP (QoS RSVP) service w values assigned to this service.
2	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the PSched (PSched) service we values assigned to this service.
3	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the RemoteAccess (Routing and contains the new index values assigned to this service.
4	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the TermService (Terminal Serv new index values assigned to this service.
5	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the MSDTC (MSDTC) service we values assigned to this service.
6	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: The Microsoft Distributed Transaction Coordinator service
7	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: MS DTC started with the following settings: Security Config Transactions = 1, Network Clients = 0, Distributed Transactions using XA Transactions =
8	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the ContentIndex (ContentIndex new index values assigned to this service.
9	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the ContentFilter (ContentFilter) index values assigned to this service.
10	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the ISAPISearch (ISAPISearch) index values assigned to this service.
11	172.19.2.29	5	2005-08-08	15:05:11	EvntSLog: Performance counters for the WmiApRpl (WmiApRpl) servi values assigned to this service.

Figure 5: A sample of the tabular view of the raw database information

The centralized syslog-ng server also hosts a Java-based graphical view of the network. This application regularly queries the database for status changes and presents these changes graphically to operators. For example, a significant status change for a particular SLAP device will cause that SLAP device's icon to "blink red". Operators can click on the icon to see the log entries that resulted in the status change. Figure 6 graphically depicts a status change, where the operator has clicked on the flashing red icon to see the relevant log information. Note: From this point, the operator can acknowledge alerts, which will turn the flashing red icon back to its normal color.



TimeStamp	...	...	Hostname	Facility	Severity	...	...	Program	
2006-06-0...	...	...	10.0.1.151	auth	alert	6	0	snort	snort[821]: [1:1100005:1] DCOM Exploit (MS03-
2006-06-0...	...	...	10.0.1.151	auth	alert	6	0	snort	snort[821]: [1:1100005:1] DCOM Exploit (MS03-
2006-06-0...	...	...	10.0.1.151	auth	alert	6	0	snort	snort[821]: [1:1101000:1] Possible dcom* c EX
2006-06-0...	...	...	10.0.1.151	auth	alert	6	0	snort	snort[821]: [1:1100005:1] DCOM Exploit (MS03-
2006-06-0...	...	...	10.0.1.151	auth	alert	6	0	snort	snort[821]: [1:1101000:1] Possible dcom* c EX
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: ip_tables: (C) 2000-2002 Netfilter core te
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY
2006-06-0...	...	...	10.0.1.151	kern	warning	3	0	kernel	kernel: IN=slap1 OUT=slap1 PHYSIN=eth1 PHY

Program	MessageData
snort	snort[842]: [1:1100005:1] DCOM Exploit (MS03-026) targeting Windows 2000 SP4 [Classification: Attempted Administrator Privilege Gain] [Priority
snort	snort[842]: [1:1101000:1] Possible dcom* c EXPLOIT ATTEMPT to 135-139 [Classification: Attempted Administrator Privilege Gain] [Priority: 1]: {T
kernel	kernel: slap1: port 2(eth1) entering disabled state
init	init: Switching to runlevel: 0
kernel	kernel: slap1: port 2(eth1) entering forwarding state
kernel	kernel: slap1: topology change detected, propagating
kernel	kernel: slap1: port 2(eth1) entering learning state
kernel	kernel: slap1: port 2(eth1) entering disabled state
kernel	kernel: slap1: port 2(eth1) entering forwarding state
kernel	kernel: slap1: topology change detected, propagating
kernel	kernel: slap1: port 2(eth1) entering learning state
kernel	kernel: slap1: port 2(eth1) entering disabled state
kernel	kernel: slap1: port 2(eth1) entering forwarding state

Severity Counts Summary

EMERG	0
ALERT	167
CRIT	2
ERROR	951
WARNING	577
NOTICE	9672
INFO	875
DEBUG	43

Start date (mm-dd-yyyy): 06-01-2006 00:00:0
End date (mm-dd-yyyy): 06-16-2006 17:00:0

Figure 6: Graphical display of an alert and corresponding log information

This page intentionally left blank.

Continuing Work

The SLAP program has attracted the attention of the NSTB program, funded by the Office of Energy Delivery and Reliability (OE) at the Department of Energy (DOE). As a result, the program has received funding in the amount of $542,000 to continue development of the SLAP.

The first step under the new programmatic direction was to rechristen the technology as the Open Architecture for Interoperability Design (OAID) program, in recognition that the best future for the SLAP lies as an industry-owned design that can be incorporated into control center software, add-on devices, and PCS field equipment alike. The SLAP will continue to serve as a reference implementation for OAID.

The OAID program calls for further technology integration (both for the design as well as the reference implementation), particularly focusing on key management for the security architecture, but also identifying the development of a configuration tool for SLAP devices as a key need. Also, the NSTB sponsorship is funding deployment of the SLAP in industry labs and operational SCADA systems for field testing, as well as the development of an industry advisory group for the transition of OAID technology.

This page intentionally left blank.

Conclusion

Although the last few years have seen the introduction of PCS security devices, none of the proposed architectures have constituted a comprehensive set of security services. The work accomplished by this project has shown several key developments, as listed below.

1. PCS systems, excepting very old legacy systems, are amenable to the application of IT security technologies;
2. A comprehensive security architecture was developed based on IT security technology; and
3. The architecture was developed as a prototype using open-source technologies.

The SLAP technology has been well-received by government and industry. In the future, we expect that it could serve as the basis for an interoperable design standard that will reduce vendor risk for PCS security development, and consequently lower the cost for application by owners of automation systems. Together, these trends may accelerate the amelioration of PCS security problems.

This page intentionally left blank.

References

[1] *A Reference Model for Control and Automation Systems in Electric Power*, Michael Berg and Jason Stamp, Sandia National Laboratories report SAND2006-742274221000C, Albuquerque, New Mexico (2005). Published in the Proceedings of the 2005 Power Systems Conference, Clemson University (March 2005).

[2] National SCADA Testbed website, The Department of Energy Office of Energy Assurance, http://www.sandia.gov/scada/National_Testbed.htm, (2006).

[3] Securing your SCADA and Industrial Control Systems, Version 1.0, Technical Support Working Group, http://www.tswg.gov/tswg/ip/SCADA_GB_Short.pdf (2005).
[Note: An unabridged version of the report is available. Contact information can be found on the TSWG website: http://www.tswg.gov/tswg/home/home.htm].

[4] Overview of Cyber Vulnerabilities, United States Computer Emergency Readiness Team, June 22, 2006, http://www.us-cert.gov/control_systems/csvuls.html

[5] *Common Vulnerabilities in Critical Infrastructure Control Systems*, Jason Stamp, John Dillinger, William Young, and Jennifer DePoy, Sandia National Laboratories report SAND2006-742274221772C, Albuquerque, New Mexico (2003). Presented at SANS SANSFIRE 2003 and National Information Assurance Leadership Conference V - (NIAL), July 14-22, 2003, Washington, DC), http://www.sandia.gov/scada/documents/031172C.pdf.

[6] National Information Systems Security Glossary (INFOSEC), National Security Telecommunications and Information Systems Security Committee, NSTISSI No. 4009, NSTISSC Secretariat (I42), National Security Agency, Ft. Meade, Maryland (September 2000).

[7] Information Security Primer, Craig Lindner, (2001) http://www.sans.org/rr/whitepapers/basics/443.php

[8] *Using Host-based Anti-virus Software on Industrial Control Systems: Integration Guidance and a Test Methodology for Assessing Performance Impacts*, Joe Falco, Steve Hurd, Dave Teuman, National Institute of Standards and Technology draft report, Gaithersburg, Maryland (2006) http://www.isd.mel.nist.gov/projects/processcontrol/AV_Guide_PCSF_Draft_Release_20060530.pdf

[9] *Adaptive Network Countermeasures*, Jamie Van Randwyk, Eric Thomas, Anthony Carathimas and Randy McClelland-Bane, Sandia National Laboratories report SAND2006-742274228624, Livermore, California (2003).

[10] *Fundamental Security Practices for Control and Automation Systems in Electric Power*, Jason Stamp, Michael Berg, Alex Berry, Raymond Parks and Bryan Smith, Albuquerque, NM (to be published)

[11] *Roadmap to Secure Control Systems in the Energy Sector*, US Departments of

Energy and Homeland Security, Natural Resources Canada (2006).

Distribution

1	MS0188	D. Chavez, LDRD Office, 1011
1	MS1221	M. Scott, 5600
1	MS0671	G. Rivord, 5610
3	MS0672	J. Stamp, 5615
1	MS0672	R. Hutchinson, 5616
1	MS0672	A. Berry, 5616
1	MS0672	A. Chavez, 5616
1	MS0672	D. Duggan, 5616
1	MS9151	L. Napolitano, 8900
1	MS9151	H. Hirano, 8960
1	MS9011	E. Talbot, 8965
3	MS9011	S. Hurd, 8965
2	MS9018	Central Technical Files, 8944
2	MS0899	Technical Library, 4536