Connecting SCADA Systems to Corporate IT Networks Using Security-Enhanced Linux

Ryan Bradetich
Schweitzer Engineering Laboratories, Inc.

Paul Oman
University of Idaho, Department of Computer Science

Connecting SCADA Systems to Corporate IT Networks Using Security-Enhanced Linux

Ryan Bradetich, *Schweitzer Engineering Laboratories, Inc.*
Paul Oman, *University of Idaho, Department of Computer Science*

Abstract—**Substation networks have traditionally been isolated from corporate Information Technology (IT) networks. Hence, the security of substation networks has depended heavily upon limited access points and the use of point-to-point Supervisory Control and Data Acquisition (SCADA) specific protocols. With the introduction of Ethernet® into substations, pressure to reduce expenses and provide Internet services to customers has many utilities connecting their substation networks and corporate IT networks despite the additional security risks. While current SCADA security literature is advocating traditional IT security safeguards, such as strong passwords, encrypted communications, and firewalls, there is no assurance that these mechanisms will provide adequate security to critical real-time control networks. Digital relays and other protection-level Intelligent Electronic Devices (IEDs) can be securely connected to SCADA systems and/or corporate IT networks via a Security-Enhanced Linux SCADA proxy that acts as a "check-valve" to allow or deny access based on preprogrammed security policies. The Security-Enhanced Linux SCADA proxy enables protection and integration engineers to meet defined or defacto security principles for network security, such as those specified in the Trusted Computer Security Evaluation Criteria (TCSEC) "Orange Book" or the newer ISO/IEC "Common Criteria." For example, the Security-Enhanced Linux SCADA proxy could be configured to allow plaintext, read-only access to some IEDs while enabling authenticated and encrypted full access to others. This paper will show how the Security-Enhanced Linux SCADA proxy can be configured to restrict data access according to company policies and/or roles.**

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) and corporate Information Technology (IT) networks have evolved independently and have historically remained isolated from each other. To reduce costs and capitalize on common standards, vendors and business managers are connecting SCADA systems with corporate IT networks. Current SCADA security literature is advocating traditional IT security solutions, such as strong passwords, encrypted communications and firewalls. No assurance exists that these mechanisms can provide adequate security for critical real-time control networks.

SCADA systems operate fundamentally different than corporate IT networks. SCADA systems manage critical infrastructure such as the transmission and distribution of electricity. Corporate IT networks manage business. Outages on the corporate IT network are generally financial and localized to a specific corporation. SCADA system outages may result in environmental damage and/or the loss of human life. Unsurprisingly, the protocols used on SCADA and corporate IT networks are also fundamentally different. SCADA protocols provide efficient, deterministic communications between devices. Corporate IT protocols generally provide reliable communication over shared communication channels.

The following three intrusions illustrate the importance of maintaining isolated SCADA systems:

- Beginning in January 2000, Vitek Boden waged a three-month war against Maroochy Water Services in Australia by dumping millions of gallons of sewage into waterways, hotel grounds and canals around the Sunshine Coast suburb [1]. Boden, a disgruntled ex-employee of the equipment supplier, then argued for a consulting job to fix the problems he had created [2]. Boden compromised the SCADA system by using a radio transmitter and acted as a fake pumping station. In addition to illustrating how SCADA systems may be vulnerable to insider attacks, this scenario also shows how attackers can use insecure wireless networks to gain access to SCADA systems.

- On January 25, 2003, the Davis-Besse nuclear power plant was infected with the MS SQL Slammer worm. Due to the infection, the Safety Parameter Display System (SPDS) was unavailable for 4 hours and 50 minutes and the plant process computer was unavailable for 6 hours and 9 minutes. The corporate firewall would have blocked the MS SQL worm infection, but a consultant had an unprotected T1 line behind the firewall [3].

- On August 21, 2006, Unit 3 of the Brown's Ferry nuclear power plant was manually shutdown after two of the recirculating pumps failed. A malfunctioning Programmable Logic Controller (PLC) caused a spike in data traffic (called a data storm). The increased data traffic caused the pumps to lock up. Conversations between the Nuclear Regulatory Commission (NRC) and the Department of Homeland Security committee staff suggested this could be an external attack [4] [5].

In this paper, we show how IT Security Solutions can be combined with a customized Security-Enhanced Linux SCADA proxy to allow secure remote read access to a SCADA system. Fig. 1 illustrates how the Security-Enhanced Linux SCADA proxy would perform a similar function to a guard device in a United States government agency network. These devices permit authorized traffic and block unauthorized traffic from crossing network boundaries with much higher assurance levels then traditional firewall solutions.
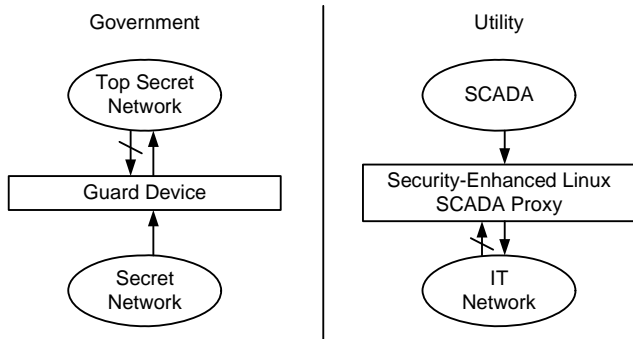
Fig. 1. Similarity: Security-Enhanced Linux SCADA Proxy vs. Government Guard Device

II. IT Security Solutions

Before evaluating the effectiveness of IT security solutions for protecting SCADA systems, one underlying assumption about corporate IT environments must be exposed: corporate IT environments implement access controls. Access to the IT network typically requires users to log in or authenticate. The user's identity determines the access the user has to various system resources (e.g., files, network shares, etc.). SCADA systems may implement some access controls for protecting device settings, but rarely provide any access controls for device operations via SCADA protocols. This means it may be possible for an intruder to operate a SCADA device (e.g., remote terminal unit or protective relay) without any form of authentication to the device.

To appreciate how access controls work, a brief introduction to the reference monitor concept is needed. The reference monitor identifies active entities (users and processes) as subjects and passive resources (files, network interfaces, etc.) as objects. The reference monitor validates accesses between subjects and objects by applying rules from a security policy. To provide effective access controls, the reference monitor must observe the following characteristics [6]:

- Tamper-proof: The reference monitor cannot be maliciously modified.
- Nonbypassable: Subjects cannot avoid the access control decisions.
- Verifiable: The reference monitor is correctly implemented and the security policy can be demonstrated.

The Trusted Computer Security Evaluation Criteria (TCSEC) defines the model implemented in corporate IT environments as Discretionary Protection (Division C in the TCSEC). The Discretionary Protection division in the TCSEC requires the system to identify and authorize users using a protected mechanism (i.e., passwords). Once the user has been identified, the reference monitor validates requests to the system objects.

With the knowledge of the implicit access controls on the corporate IT network, it is possible to examine how these IT security solutions are used differently on each network.

A. Strong Passwords

This paper assumes the corporate IT environment uses a single-factor, password-based authentication model. While two-factor and three-factor authentication models are becoming more popular in corporate IT environments, they are not universally adopted and are rare in SCADA environments. The single-factor authentication model requires the knowledge of a secret. The account owner shares a secret (the password) with the system. This model assumes any user able to provide the correct password must be the account owner and is authenticated as the user.

The single-factor authentication model has a major design flaw. It cannot distinguish account owners when the password is shared or otherwise compromised. Most IT systems discourage sharing passwords, but two common techniques intruders can use for password guessing are dictionary and brute-force attacks. Dictionary attacks rely upon the knowledge that people will tend to pick easy to remember passwords. Brute-force password attacks generate and test every possible password combination.

To help mitigate the single-factor authentication risks, many corporate IT computers are configured to require periodic password changes and the enforcement of hard passwords. Periodic password changes reduce the time a password is valid, thus reducing the risk an intruder can successfully guess the password. Strong passwords expand the search space an intruder needs to search, increasing the average time the intruder needs to successfully guess the password. Oman, Schweitzer, and Frincke illustrate the importance of using hard passwords by comparing the time difference between dictionary and brute-force password guessing attacks in a typical substation controller [7].

Password usage in the SCADA environment is significantly different from the use of passwords in the corporate IT environment. The operation of SCADA devices, such as the closing of valves or the opening of circuit breakers are typically issued as commands via a SCADA protocol. Since most SCADA protocols do not support user authentication, it is rare to find password protection on these operations. Rather, passwords typically protect settings in the various SCADA devices. Unlike the corporate IT environment, most SCADA devices cannot be configured to require periodic password changes or enforce the usage of hard passwords. In fact, many SCADA devices may not even be capable of hard passwords since they may require passwords entered on a numeric keypad.

B. Encrypted Communications

Encryption is commonly used to provide confidentiality and/or authentication. Encryption permits senders and receivers to communicate with each other while preventing outsiders from listening to the conversation. Encryption can also be used for authentication purposes by producing digital signatures. These digital signatures can then be independently verified to ensure the identity of the individual or remote system.

When applied to corporate IT networks, Virtual Private Networks (VPNs) routinely communicate confidential information between remote offices and business partners via the public Internet. VPNs consist of two or more VPN end-point devices to create an encrypted virtual network circuit. To pre-

vent outsiders from gaining access to the confidential information, all the data passing through the virtual circuit is encrypted and decrypted by the VPN end-points.

Public Key Infrastructure (PKI) systems are commonly used in corporate IT networks for authenticating both users and servers. PKI systems aid in authentication by having a trusted third party independently verify the identity of the user or server. Once verified, the trusted third party digitally signs a certificate proving the verification occurred. Two common examples of PKI systems in corporate IT networks are the Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) protocol and the smart card authentication in newer Microsoft Windows operating systems.

Encryption is primarily used in SCADA systems to protect communications from outside listeners. SCADA systems routinely use VPN and encrypted serial transceivers to ensure privacy over long communication lines. Encryption is not commonly used for authentication purposes in SCADA systems.

The underlying assumption of access controls on the corporate IT network provides a false sense of security when using encryption in SCADA systems. In the corporate IT model, it is implicitly assumed the intruder will still need to authenticate to obtain access to the resources. In the SCADA model, this assumption is not valid for the majority of the SCADA protocols available today. It is very likely an intruder could issue a DNP command to trip an electric circuit breaker through a VPN connection without authenticating to the IED device. It is very unlikely an intruder could issue a command to delete all the data from a corporate IT database without authenticating to the database.

C. Firewalls

Firewalls implement a specific security policy to protect resources on computer servers and networks. Firewalls typically provide perimeter defense by applying access controls to all network traffic traveling between networks. A simple, corporate IT network firewall is similar to the stonewall surrounding a medieval village. The firewall serves as a protective barrier for the valuable contents of the corporate IT systems. Once the firewall has been penetrated, it provides little protection against an intruder monitoring or altering data inside the protected network [8]. Packet-filtering and proxy firewalls are two common firewall types used in corporate IT environments.

The packet-filtering firewall inspects all packets going through the firewall. Access controls are used to determine if the packet should be discarded or forwarded. Most packet-filtering firewalls operate at Open Systems Interconnection (OSI) layer 3 (network) and layer 4 (transport), and inspect the information in the IP, TCP, and UDP headers. Packet-filtering firewalls typically look at source and destination addresses, source and destination ports, inbound or outbound packet, which interface the packet is arriving or leaving on, network header options, and transport layer type [9].

Two improvements on the simple packet-filtering firewall include stateful firewalls and application firewalls. Stateful packet-filter firewalls store state information about each connection. This allows the firewall to implement more complex rules such as allowing return packets to a protected device when the connection originated from the protected network. If the protected device did not initially contact the source host for the packet, the firewall will discard the packet. Application firewalls have knowledge about specific applications or protocols. This knowledge allows the firewall to make decisions based on information present in the application layer. The firewall makes these decisions by looking at the packet's contents using "deep packet inspection." Application firewalls are still a developing market and are not commonly available for SCADA protocols [9].

The proxy firewall intercepts connection requests to the protected networks and, if the connection attempt is authorized, connects to the destination device on behalf of the user. One distinguishing feature of the proxy firewall versus the packet-filter firewall is that the proxy firewall can require the user to authenticate with the proxy before authorizing the connection. Proxy firewalls can operate at the networking and applications layers. This can offer some significant advantages, such as prohibiting all inbound HTTPS traffic containing scripts [9].

The use of firewalls to provide one-way communication between SCADA and corporate IT networks is fundamentally flawed. Firewalls operate by first receiving data, then checking access controls, and finally discarding or forwarding the data. This is unacceptable for SCADA systems because (1) firewalls can forward malicious data (e.g. breaker trip commands) into the SCADA environment and (2) the corporate IT network can interfere with SCADA systems by altering the network load (e.g., denial-of-service attacks or cause network load-sensitive SCADA devices to misoperate). The access control model proposed in this paper replaces the traditional IT firewalls when connecting SCADA and corporate IT networks. The Security-Enhanced Linux access control model fundamentally operates differently from the firewalls described thus far. The Security-Enhanced Linux SCADA proxy never forwards data from the corporate IT network to the SCADA system. Instead, data from the SCADA system is periodically collected, cached, and provided to the corporate IT users upon request. This model eliminates the possibility of malicious data coming from the corporate IT networks and entering the SCADA system. Further, predictable traffic patterns and the use of cached data prevent the corporate IT network from interfering with SCADA environment via denial-of-service attacks or "data storm" failures.

III. SECURITY-ENHANCED LINUX SCADA PROXY

The IT security model is only marginally sufficient for protecting corporate IT networks from intrusions. Applying a flawed model to critical control networks with different underlying assumptions and characteristics is almost inviting security incidents and SCADA outages. A different model for connecting SCADA systems to other networks is needed. This model needs to permit the transmission of data from the

SCADA system to other networks, but prohibit communication and interference from the other network.

The United States government has a similar problem in dealing with classified networks (e.g., top-secret, secret, confidential, unclassified). These networks are generally isolated from each other, but when communication does occur between different classified networks, the communication must occur through a guard device. The guard device provides assurance the remote network is not accessible to an intruder, even when the guard device has been compromised.

The Security-Enhanced Linux SCADA proxy was modeled after a very basic guard device configuration. The primary goal of the Security-Enhanced Linux SCADA proxy was to act as a check-valve and allow specific SCADA information to leave the SCADA environment and prevent all corporate IT traffic from entering or interfering with the SCADA system.

The Security-Enhanced Linux SCADA proxy enforces this one-way communication by logically separating the running processes into 13 application domains:

- Kernel Domain
- Syslog Domain
- Netadmin Domain
- Init Domain
- Firewall Domain
- Web Domain
- Web Protection Domain
- SCADA Protection Domain
- Web Meter Domain
- SCADA Meter Domain
- Untrusted Simple Mail Transfer Protocol (SMTP) Domain
- SMTP Proxy Domain
- Trusted SMTP Domain

Each application domain is responsible for a specific function and has been granted access to the necessary resources to perform this function. The application domains have been logically organized into Core Domains, SCADA Data Proxy Domains, and SMTP Proxy Domains. This logical grouping and communication path between the logical domains is shown in Fig. 2.
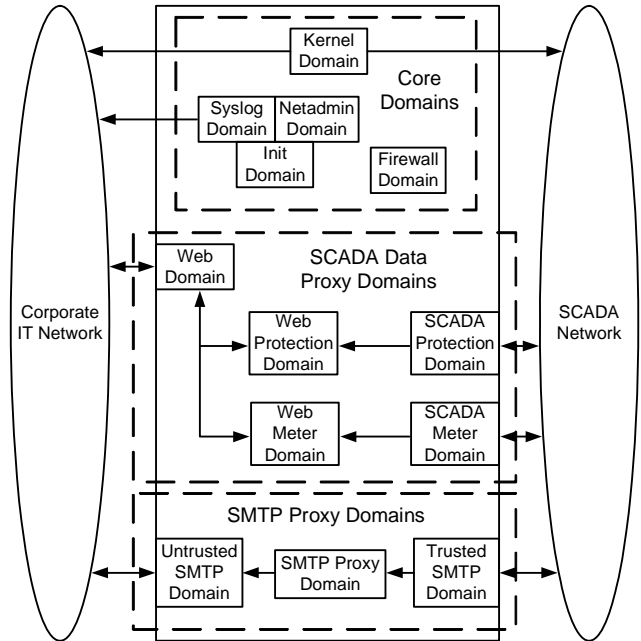

Fig. 2. Security-Enhanced Linux SCADA Proxy—Application Domains and Communication Paths

A. Core Domains

The Core domains are composed of five application domains:

- Kernel Domain
- Syslog Domain
- Netadmin Domain
- Init Domain
- Firewall Domain

These application domains are responsible for the base functionality provided by the Linux operating system. The Kernel domain is responsible for controlling interactions between the Linux kernel and user-space applications. The Kernel domain is the only application domain that, if compromised, an intruder could circumvent the protection offered by the Security-Enhanced Linux SCADA proxy. Research is still on-going to determine if this weakness can be removed in future solutions. The Syslog domain provides a one-way communication path for sending system log messages to a centralized logging server on the corporate IT network. The Netadmin and Firewall domains are responsible for configuring the network interfaces and initializing the packet-filter based firewall built into the Linux kernel. Finally, the Init domain is responsible for application and system startup.

B. SCADA Data Proxy Domains

The SCADA data proxy collects data from devices on one network and provides this data to users or devices on different networks. This model is intentionally generic and can be applied to many different applications. One implementation could provide an adapter or bump-in-the-wire solution for connecting IEDs directly to corporate IT networks. A different implementation could provide separate access controls for utilities accessing shared resources in a substation.

The implementation used for this paper collects data from both a protective relay and a revenue meter. This data is then cached locally on the Security-Enhanced Linux SCADA proxy. A web-portal provides authorized corporate IT users read-only access to this cached SCADA data. This cache-and-collect model provides a high-assurance, one-way communication path, prevents interference from the corporate IT network, and maintains a predictable load on the SCADA system. Additionally, this implementation provides protection for other SCADA devices from an intrusion originating from the SCADA system.

The SCADA proxy is composed of five application domains:

- SCADA Protection Domain
- SCADA Meter Domain
- Web Domain
- Web Protection Domain
- Web Meter Domain

The SCADA Protection Domain and the SCADA Meter Domain are responsible for collecting data from the protective relay and revenue meter. These collection domains communicate with a specific IED using SCADA protocols. This use of SCADA protocols allows the Security-Enhanced Linux SCADA proxy to appear as a SCADA device and allows control engineers to provide a known and predictable load on the SCADA system. Once the data has been collected from the IED, it is stored locally on the Security-Enhanced Linux SCADA proxy. Each SCADA collection domain is permitted to create, read, write, and delete the cached information for the IED it is responsible for. Access is prohibited to other IED's cached data.

The Web Domain, Web Protection Domain, and Web Meter Domain are responsible for providing read-only access to the cached SCADA data to authorized users on the corporate IT network. The Web domain is responsible for accepting encrypted web connections and authorizing corporate IT users. To provide an extra layer of security against intruders compromising the Web domain, this domain does not have direct access to any of the cached SCADA data.

When requested by authorized users, Common Gateway Interface (CGI) executables are launched from the Web Domain into either the Web Protection Domain or the Web Meter Domain to provide read-only access to the appropriate subset of the cached SCADA data.

Fig. 3 shows a pictorial representation of the domains, types, and communication paths used by the Security-Enhanced Linux SCADA proxy to enforce a one-way communication path for the SCADA data proxy.
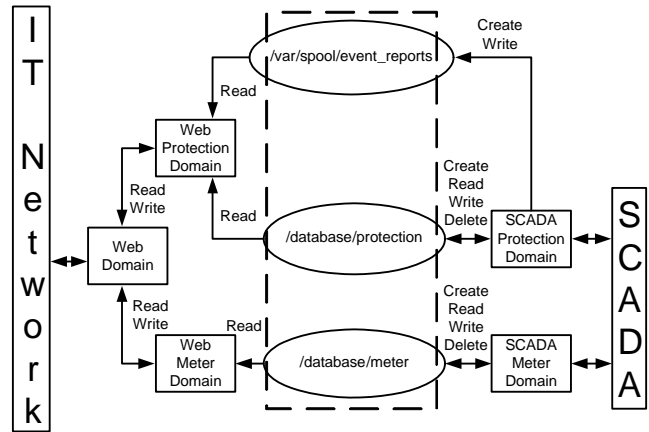

Fig. 3.   SCADA Data Proxy—One-Way Communication Diagram

The one-way communication is achieved by only allowing the SCADA protection and SCADA meter domains to create, read, write, and delete content in the appropriate SCADA data caches; while the appropriate web CGI program has read-only access to the cached SCADA data.

C. SMTP Proxy Domains

The purpose of the SMTP proxy is to provide a one-way communication path for email leaving the SCADA system. Ensuring this one-way communication path is challenging because SMTP uses the store-and-forward model along with stringent protocol specifications to ensure a reliable communication mechanism. In addition to the underlying TCP protocol requiring two-way communication, the SMTP protocol itself requires two-way messaging.

The SMTP proxy addresses this two-way communication problem by separating the SMTP transfer functionality into three application domains:

- Trusted SMTP Domain
- SMTP Proxy Domain
- Untrusted SMTP Domain

The Trusted SMTP domain provides the standard SMTP service to proxy email outside the SCADA system. The SMTP proxy permits the required two-way communication between the SCADA SMTP clients and the Trusted SMTP domain. After the email has been successfully stored on the non-volatile media in the /var/spool/postfix directory, postfix will attempt to deliver the email to the next destination through a custom postfix pipe delivery agent, which is also part of the Trusted SMTP domain. This custom delivery agent writes the email message to the /var/spool/smtp directory. To ensure the one-way communication path, the custom delivery agent only has create, write, and rename permissions to the /var/spool/smtp directory. The SMTP Proxy domain provides a completely independent application which periodically scans the /var/spool/smtp directory for new email messages. When a new email message is found, the SMTP proxy process spawns a new SMTP client process in the Untrusted SMTP domain to complete the email message delivery to the corporate IT mail server. To ensure the one-way communication path, the SMTP Proxy domain only has read and delete privileges to the /var/spool/smtp directory. The Untrusted SMTP domain is

only allowed to read messages from the /var/spool/smtp directory.

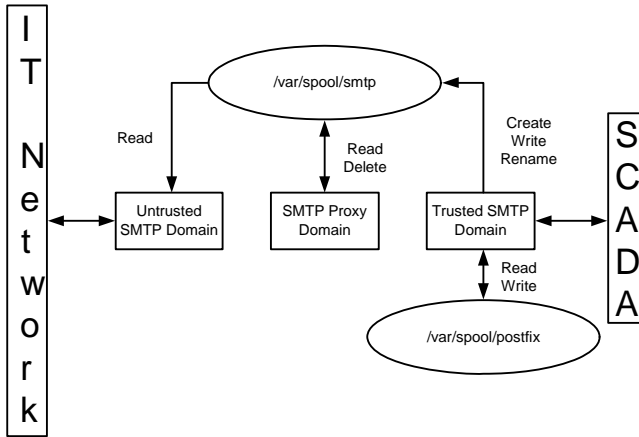Fig. 4 shows a pictorial representation of the domains, types, and communication paths.

Fig. 4.   SMTP Proxy—One-Way Communication Diagram

D. Adding Assurance with Security-Enhanced Linux

In a perfect world, where software flaws and malicious intruders do not exist, the Security-Enhanced Linux SCADA proxy could provide improved protection for SCADA systems using just the standard Linux discretionary access controls. Unfortunately, both software flaws and malicious intruders are prevalent. An additional level of access controls beyond discretionary access controls are needed because the discretionary access control model suffers from three flaws that make it unacceptable for use in the Security-Enhanced Linux SCADA proxy:

1. An insider is an intruder attacking a computer or network to which they have legitimate access. The discretionary access control model allows object owners to alter the security policy by modifying object attributes (e.g., file permissions). Thus, an insider can either create a denial-of-service attack or share data with unauthorized users by altering the security policy.

2. The discretionary access control model cannot distinguish between interactive and non-interactive process when performing access control checks. To illustrate this point, assume there are two processes running on the system as the rbrad user: a web server daemon (non-interactive) and a shell program (interactive process). The discretionary access control model cannot enforce a security policy where only the interactive shell has the ability to change the password on the *rbrad* account. Thus, if the non-interactive daemon is somehow corrupted, it can change object permissions surreptitiously (e.g., buffer overflows).

3. Discretionary access control systems typically have a super-user account used for system administrative tasks. This account circumvents the access control checks in the reference monitor to resolve access control issues on the system. Intruders can now either find flaws in the reference monitor or compromise the super-user account to obtain full access to the system.

Security-Enhanced Linux is an optional mandatory access control implementation provided in the Linux kernel. The mandatory access control model was developed throughout the 1970s and 1980s to address issues with the discretionary access control model and problems with flawed and malicious software [6]. This access control model is defined as Mandatory Protection (Division B in the TCSEC).

In addition to implementing all the requirements for the Discretionary Protection division, the Mandatory Protection division requires all processes, resources, and data to be labeled. These labels are used by the mandatory access control model to implement an organizational security policy, which cannot be altered by users or programs [6].

Security-Enhanced Linux is based on the Type Enforcement (TE) mandatory access control model, which, was designed to provide integrity for critical government computers operating at the A1 level (Verified Protection) of the TCSEC [10]. Security-Enhanced Linux implements additional, fine-grained access controls by separating critical components into application domains. All subjects (processes) are assigned a domain label and are confined to run in a specific application domain. Objects (files, directories, network interfaces, etc.) are also assigned a label. Each object type defines a set of available operations. For example, the file object type may define create, delete, read, write, and execute operations, while, the socket object type may define the create, delete, connect, and bind operations. Relationships between the subject labels and object labels are defined in the Security-Enhanced Linux security policy. The reference monitor uses the Security-Enhanced Linux security policy file to determine if the subject has the requested access to the object.

Applications domains in the type enforcement model are designed to separate the operating system from applications and different applications from each other [10]. If an intruder compromises a domain, the intruder is limited to privileges of that domain. To compromise other domains, the intruder would need to bypass both the discretionary access and type enforcement controls.

To test that the Security-Enhanced Linux security policy properly protects the SCADA system from intruders from the corporate IT network, an intrusion exploit suite of tools (rootkit.cgi) was installed and executed on the Security-Enhanced Linux SCADA proxy. The rootkit.cgi exploit suite was given Set User ID (suid) privileges and was run in the web application domain to simulate an intruder that has successfully compromised the web server and obtained super-user privileges. This test ensures the intruder cannot violate the one-way communication path shown in Fig. 3 by preventing the intruder from making modifications to the file system objects. The rootkit.cgi performs a complete file system scan and attempts to perform the following actions on each file system object:

1. Create a new file in each directory.
2. Open the file system object with read-only permissions.
3. Open the file system object with append permissions.
4. Open the file system object with write-only permissions (truncates the file).
5. Remove the file system object.

Running this test on a system protected only with discretionary access controls would remove all files, leaving only directories and symbolic links, but does nothing on the Security-Enhanced Linux SCADA proxy running mandatory access controls. This test demonstrated the effectiveness of the Security-Enhanced Linux SCADA proxy against a privileged intruder in the web domain. In addition to still having a functional system, this test validated the Security-Enhanced Linux security policy by prohibiting the intruder from altering the website, modifying the cached SCADA data, and communicating with serial-connected SCADA devices.

IV. CONCLUSIONS

After exploring and evaluating solutions to problems with connecting SCADA systems to corporate IT networks, the Security-Enhanced Linux SCADA proxy proof-of-concept was developed to act as a high-assurance, one-way communication path for data leaving the SCADA system. The SCADA Data Proxy provides data from SCADA devices to corporate IT users, implements a collect-and-cache model to prevent the corporate IT network from interfering with SCADA devices, and maintains a predictable load on the SCADA system. SCADA devices were separated into individual domains to prevent intruders from the SCADA environment from using the Security-Enhanced Linux SCADA proxy as a platform for attacking other SCADA devices. The web interface provided a web-portal to the corporate IT users and restricted access to the data based on the user's role. The SMTP proxy was included to show how a two-way protocol could be isolated into discrete actions to enforce the one-way communication requirements. The SMTP proxy was broken into three domains: Trusted SMTP, SMTP Proxy, and Untrusted SMTP. By locally delivering email to files and using Security-Enhanced Linux to control access to the directory and files, the Security-Enhanced Linux SCADA proxy was able to provide a one-way communication path. To ensure that the Security-Enhanced Linux SCADA proxy protected the SCADA system when compromised by an intruder with root privileges, a rootkit.cgi was built to simulate the attack. The Security-Enhanced Linux mandatory access controls blocked the attack and warned of the attack by sending syslog messages to the central syslog server.

V. REFERENCES

[1] M. Crawford. (2006, Feb.). Utility Hack Led to Security Overhaul. *ComputerWorld Security.* [Online]. Available: http://www.computerworld.com/securitytopics/security/story/0,10801,108735,00.html

[2] S. Mustard. (2007). Security of Distributed Control Systems: The Concern Increases. *The Institute of Engineering and Technology.* [Online]. Available: http://www.iee.org/OnComms/sector/computing/Articles/Object/0482B3C1-D0B1-80C6-57EA91E4FB429C23

[3] W. Beckner. (2003, Aug.). NRC Information Notice 2003-14: Potential Vulnerability of Plant Computer Network to Worm Infection. *United States Nuclear Regulatory Commission.* [Online]. Available: http://www.nrc.gov/reading-rm/doc-collections/gen-comm/info-notices/2003/in200314.pdf

[4] R. Lemos. (2007, May). "Data Storm" Blamed for Nuclear-Plant Shutdown. *SecurityFocus.* [Online]. Available: http://www.securityfocus.com/news/11465

[5] M. Rasmussen. (2006, Aug.). NRC: Event Notification Report for August 21, 2006. *United States Nuclear Regulatory Commission.* [Online]. Available: http://www.nrc.gov/reading-rm/doc-collections/event-status/event/2006/20060821en.html#en42787

[6] F. Miller, K. MacMillan, and D. Caplan, *SELinux By Example.* Prentice Hall, 2007.

[7] P. Oman, E. Schweitzer, and D. Frincke, "Concerns About Intrusions into Remotely Accessible Substation Controllers and SCADA Systems," in 2000 27th *Annual Western Protective Relay Conference Proceedings.* Also presented at the Monterrey Symposium on Electric Systems Protection, Monterrey, Mexico, 2000, and at the Georgia Tech Protective Relaying Conference, Atlanta, Georgia, 2001.

[8] G. Leischner and C. Tews, "Security Through VLAN Segmentation," presented at the 33rd Annual Western Protective Relay Conference, Spokane, Washington, 2006.

[9] E. Byres, J. Karsch, and J. Carter. (2005, Feb.). NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks. *National Infrastructure Security Co-Ordination Centre.* [Online]. Available: http://www.cpni.gov.uk/docs/re-20050223-00157.pdf

[10] Secure Computing. (2007). Sidewinder Type Enforcement Technology. *Secure Computing.* [Online]. Available: http://www.securecomputing.com/index.cfm?skey=738

VI. BIOGRAPHY

Ryan Bradetich received his BSCS in 1997 and his MSCS in 2007 from the University of Idaho. He is a lead database developer at Schweitzer Engineering Laboratories, Inc. in Pullman, WA. Ryan is currently working on automation products used in electric utility substations. Prior to joining SEL, he worked at Hewlett-Packard on the security team responsible for auditing and reporting the security status for approximately 20,000 UNIX and Windows® systems.

Dr. Paul W. Oman is a Professor of Computer Science at the University of Idaho. He is currently working on secure communications and critical infrastructure protection with grants from NSF, NIATT, and DARPA. From 2000 to 2002, he served as a Senior Research Engineer at Schweitzer Engineering Laboratories, specializing in digital equipment for electric power system protection. Before joining SEL, he was Chair of the CS Department and held the distinction of Hewlett-Packard Engineering Chair for a period of seven years. He is a Senior Member of the IEEE.