Chapter 12

INTRUSION DETECTION AND EVENT MONITORING IN SCADA NETWORKS

Paul Oman and Matthew Phillips

Abstract This paper describes the implementation of a customized intrusion detection and event monitoring system for a SCADA/sensor testbed. The system raises alerts upon detecting potential unauthorized access and changes in device settings. By markedly increasing the logging of critical network events, the system shows dramatic improvements in both the security and overall auditing capabilities. In addition to its role in securing SCADA networks, the system assists operators in identifying common configuration errors.

Keywords: Intrusion detection, real-time monitoring, SCADA networks

1. Introduction

Power system control was once a laborious process. Prior to the use of digital control equipment and communications networks, engineers had to travel to each substation to view system conditions and make changes. Technological advances enabled engineers to monitor their systems from a central location, controlling dozens of substations from a single terminal [8]. Further advances now permit engineers to control their systems – even from home – using the Internet, telephone system and wireless networks [2].

When control systems were stand-alone, devices were required to meet strict standards on operating temperatures, electrical disturbances and other environmental concerns. The operating environment has changed. In addition to meeting the harsh realities of an industrial environment, engineers must now account for new "disturbances" – electronic attacks.

Process control systems are very heterogeneous environments. A power substation may have devices from a dozen different manufacturers. Some devices may communicate serially or via proprietary protocols on proprietary cabling, others may use Ethernet, and still others tunneling protocols over Ethernet. Some devices may be 20 years old, while others are brand new.

Oman, P. and Phillips, M., 2008, in IFIP International Federation for Information Processing, Volume 253, Critical Infrastructure Protection, eds. E. Goetz and S. Shenoi; (Boston: Springer), pp. 161–173.

Process control systems are built to operate in high stress, time-sensitive environments. The devices are simple and dedicated to performing their limited tasks well. Therefore, most devices do not have the memory, processing power and bandwidth required to perform security functions. Real-time control systems have several additional limitations, including:

- Weak authentication mechanisms that do not differentiate between human users.
- No privilege separation or user account management to control access (e.g., one account, one password).
- Most devices do not record login attempts (e.g., success, failure and number of attempts).
- Most devices cannot run user programs; they can only perform simple logic operations.
- Many users do not change the factory default settings of devices.
- Many control networks are not designed with cyber security in mind.
- Proprietary protocols slow the integration of security tools in control networks.
- Overall lack of monitoring and auditing (e.g., tracking changes to settings and firmware upgrades).
- Devices are notoriously difficult to set up and are typically configured once and left alone.
- Heterogeneous control networks with components varying in age and capabilities require singular attention to secure, making broad adoption unaffordable.

These factors severely hamper efforts to secure control systems [1, 9]. Fortunately, the solutions are well-known in the information technology field [5]. Indeed, many security solutions can be realized using existing technology at a reasonable cost.

We have identified common security weaknesses in automated process control systems, with particular attention to remotely-accessible power substations [3, 4], and have created a model SCADA/sensor testbed for experimentation. This paper describes the implementation of a customized intrusion detection and event monitoring system for the testbed. The system raises alerts upon detecting potential unauthorized access and changes in device settings. It is useful for securing SCADA networks as well as assisting operators in identifying erroneous or malicious settings on SCADA devices.

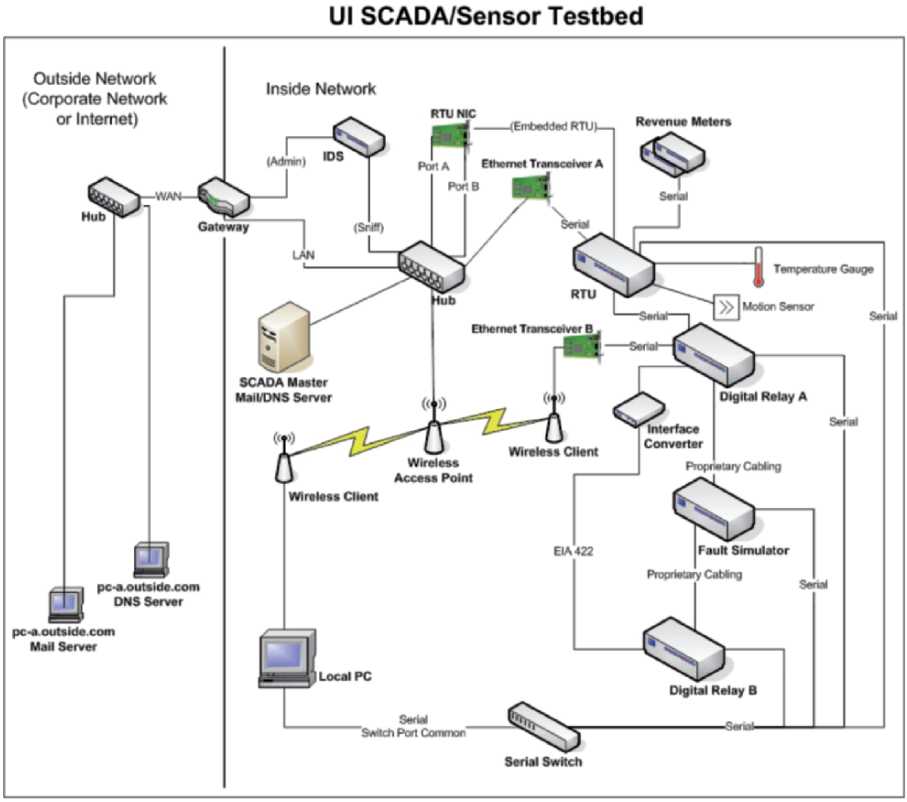

Figure 1. SCADA/sensor system testbed.

2. SCADA/Sensor System Testbed

Our SCADA/sensor system testbed was created to provide a learning and research environment for efforts related to SCADA security and survivability. The testbed leverages facilities at the University of Idaho's Electrical Engineering Power Laboratory, a fully functioning high-voltage facility [10, 11].

A schematic diagram of the testbed is presented in Figure 1. The testbed incorporates a communications system, sensor system, digital fault simulator and a priority messaging system. The communication system includes a wired Ethernet network, which simulates Internet or corporate LAN traffic, and an 802.11b wireless network. These networks connect a substation communications processor to the SCADA master unit and other computers to enable remote access. The communications processor is a microprocessor-based device that replaces the traditional (and archaic) remote terminal unit (RTU) still found in many SCADA systems. It is logically programmable and serves as a data collection and communications hub with connections to the sensor system and protective relay equipment. The wireless component of the communications

system consists of two wireless bridges configured to communicate via a point to multi-point topology. Protective relays are used to monitor the power system, control circuit breakers and report faults. The testbed incorporates motion and temperature sensors that raise various alarm conditions.

The power laboratory includes electrical machinery with integral horsepower motor-generator sets ranging in size from 5 to 20 HP. A mix of AC and DC machines permits flexible experimentation with active loads. The largest is a 20 HP synchronous generator used to protect generators from internal faults. This machine, which has been modified to support development and testing schemes, is connected to the SCADA/sensor systems via power quality measurement equipment. Supply capability includes: 240V three phase AC at 115A, 120V three phase AC at 150A, 120V at 400A DC, and 240V at 180A DC. Each supply is fed at 480V three phase AC via transformers housed in the laboratory. DC is generated by motor-generator sets.

The laboratory also incorporates a transient network analyzer, which can be configured to have four transmission line segments for modeling a transmission system. Full instrumentation is available for SCADA and power system protection; this facilitates a wide range of experimentation related to protecting power systems. The controls for the prime movers on the system are adjustable, allowing it to reproduce dynamic oscillations on a power grid and to demonstrate how changes in SCADA control settings can impact its behavior. The system can also be used for modeling and testing custom electronic power controllers. Central to the ability to perform analysis of specific transient scenarios is the implementation of a computer-controlled fault generator. The fault generator enables complex multiple and progressive faults to be modeled, making real-time voltage and current behavior during these events available for analysis. The laboratory incorporates mechanical circuit breakers controlled by commercial protective relays.

3. Research Objectives

A network intrusion detection system acts as an eavesdropping tool, listening on a network for different types of traffic and payload data. Such a tool could noticeably improve security in a SCADA network. SCADA networks also need tools that remotely track changes to device configurations and settings. Monitoring network traffic and auditing network device settings provide the basis for intrusion detection in IT networks; they are just as effective in SCADA networks [6, 10, 11].

Our research had three objectives. First, we wanted to better secure the communication systems of SCADA networks by monitoring for commands that could adversely impact the reliable operation of these networks. Second, we wanted to better monitor the settings on SCADA devices by using an automated technique for gathering settings and comparing them with known (working) values. Third, we wanted to use existing technologies in an extensible, cost-effective approach to improving intrusion detection and event monitoring in SCADA networks.

Industrial control networks severely underreport many important details regarding system access. Therefore, any effort to detect intrusions must involve the observation and recording of important network events. These events include:

- Login attempts on a network device, including:
  - Time of day
  - Origin and destination IP addresses of the attempt
  - Whether the attempt succeeds or fails
  - Frequency of attempts over a given time interval
- Major SCADA-specific commands, including:
  - Commands to view or set passwords
  - Commands to upload new firmware
  - Commands to show or change settings
  - Attempts to upgrade user privileges

Our intent was to incorporate intrusion detection and event monitoring in the testbed. Due to the critical nature of the work performed by SCADA devices, it is important to record even legitimate access attempts. Moreover, research shows that many errors can be attributed to mistakes made by SCADA operators; it is logical to provide services to reduce human error and mitigate any adverse effects.

4. Prototype System

The automated gathering and comparison of device settings over time can be very useful to SCADA operators, who typically rely on personal notes and reminders about which settings were changed and when. Because `telnet` is the most common means for connecting to SCADA devices, we chose to automate this process using the Perl programming language and its Expect module that automates interactions with other programs. This combination has simplified the automation of terminal connections with various SCADA devices. Moreover, it readily supports secure connection protocols like SSL and `ssh`.

Figure 2 presents a logical diagram of the intrusion detection and event monitoring system. To complement settings gathering and command logging, we added a customized uptime measurement component to the testbed. Using `ping`, `telnet`, Expect and a database backend, we were able to graphically represent the uptimes of each SCADA device over day-, fortnight- and month-long periods. This proved to be very effective in identifying faulty devices and network paths, especially involving devices that are seldom used but that are expected to be reliable. Network connectivity was tested for all SCADA devices and the mean time to repair (MTTR) was computed for each device.

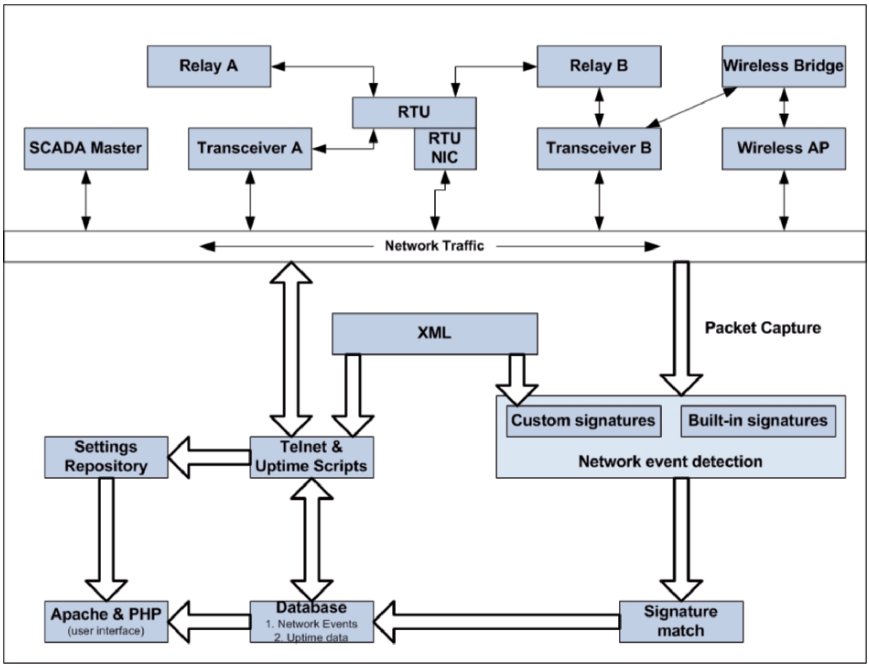

Figure 2. Logical diagram of event monitoring flow and SCADA testbed components.

4.1 Intrusion Signature Generation

Details about each SCADA device in the testbed are expressed using XML. XML provides a standard way to describe diverse SCADA devices. Moreover, the XML format is very expressive and highly extensible.

Details stored about each device include its IP address, `telnet` port, legal commands for the device, whether or not to create intrusion signatures for specific commands, and whether or not to issue a certain command during the automated process of retrieving settings. Table 1 shows a portion of the XML profile for the RTU.

Table 2 lists many of the legal commands available on the RTU. Each command has an entry in the RTU's XML profile. A Perl program parses the XML profile and creates a Snort IDS signature [7] for legal commands on the RTU in order to monitor normal operations. Two automatically-generated signatures are shown in Table 3.

Since there well over 100 signatures, it is beneficial to have a mechanism that can automatically generate IDS signatures. However, not all signatures can be created in this manner. Failed password attempts, for example, require pattern matching on the RTU's failed response to a bad login attempt. In this case, a packet sniffer is used to determine the response and a customized signature is created to detect login failures, which are then graphed over various

Table 1. XML profile for the RTU.

```
<?xml version="1.0"?>
<device>
    <device_name>Remote Terminal Unit</device_name>
    <ip>192.168.0.17</ip>
    <telnet_port>23</telnet_port>
    <admin_port>1024</admin_port>
    <description>This device serves as the communications
    processor in the testbed.</description>
    <level1_user>ACCESS1</level1_user>
    <level1_pass>PASS_1</level1_pass>
    <level2_user>ACCESS2</level2_user>
    <level2_pass>PASS_2</level2_pass>
    <cmd>
        <name>ID</name>
        <description>SETTINGS -- Show port settings for info
        on connected devices.</description>
        <automate>no</automate>
    </cmd> . . . . . .
```

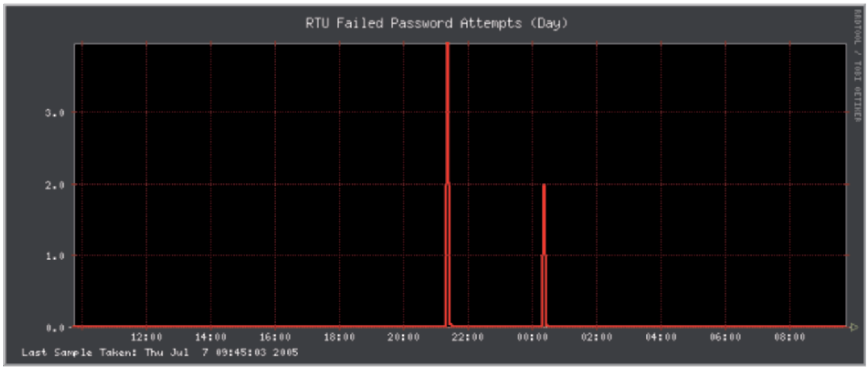

Figure 3. Graph of failed login attempts over a 24-hour period.

time periods (Figure 3). Thus, network events are detected (and subsequently graphed) using automatically-generated signatures or customized signatures for failed login attempts and other complex events. In the near future, signatures will be generated for all the devices listed in Table 4.

4.2 Monitoring Settings

The second component of our system involves monitoring changes to device settings, including changes made at the local terminal and those performed

Table 2. Common commands used in the testbed.

Command	Description
BROADCAST	Communicate with all IEDs
CLEAR	Clear information from a memory area
DNP	View DNP data and settings
MODBUS	View MODBUS data and settings
DATE	View or change the date
HELP	Provide information on available commands
ID	Display device identification information
CLOCK	Force time update using IRIG output
PORT	Provide direct access to a port
ACCESS	Change access level to Level 1
2ACCESS	Change access level to Level 2
QUIT	Revert to access Level 0
SETTINGS	Show all device settings
STATUS	Display status and configuration information
TIME	View or change the time
VIEW	View information from the database
WHO	Show directly connected devices
COPY	Copy settings between ports
LOAD	Initiate firmware upgrade sequence
PASSWORD	View or change passwords
PING	Ping a network device
FTP	FTP metering data from a device

Table 3. Signatures for ACCESS and 2ACCESS commands.

```
alert tcp $HOME_NET any -> $RTU $RTU_PORT
 (msg:"RTU 2ACCESS - Change access level to access Level 2";
 pcre:"/\b2AC/i"; session: printable sid:1200014 rev: 10;)

alert tcp $HOME_NET any -> $RTU $RTU_PORT
 (msg:"RTU ACCESS - Change access level to access Level 1";
 pcre:"/\bACC/i"; session: printable sid:1200015 rev: 10;)
```

over the network. To implement this functionality, a single settings repository is maintained for the SCADA testbed; each device has one or more baseline settings files in the repository. Successive settings are compared against the baseline settings to determine what changes have been made. It is important to know when the settings are changed because a network monitoring device cannot detect changes made from the local terminal. Monitoring settings in this manner implies that the baseline is known to be correct. Therefore, the baseline should be created before the system is brought online. Also, baseline

Table 4. Testbed devices.

Device	IP Address
RTU Network Card	192.168.0.17
Transceiver A	192.168.0.11
Transceiver B	192.168.0.12
Digital Relay A	Accessible via RTU
Digital Relay B	Accessible via RTU
Wireless AP	192.168.0.227
Wireless Bridge	192.168.0.225
Wireless Client	192.168.0.14
SCADA-MASTER	192.168.0.140
Gateway	192.168.0.1

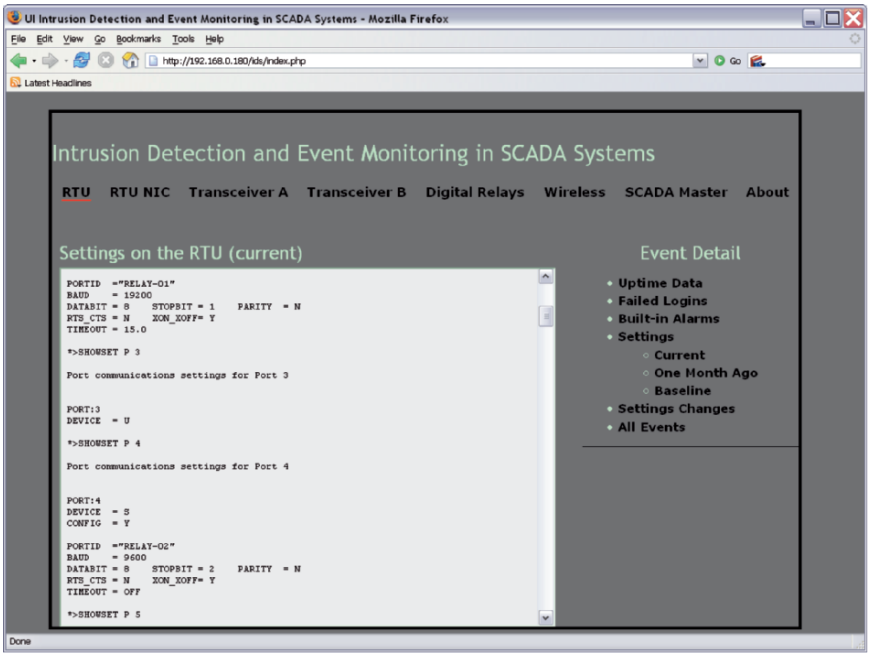

Figure 4. Screenshot of RTU settings after automated retrieval.

data should be protected from unauthorized access and modification. Figure 4 shows a screenshot of the settings recovered from the RTU.

Note that it may be infeasible to monitor every segment of a SCADA network, which is often the case when a wireless network is used to connect remote devices and/or substations. Fortunately, proper network design at the outset can alleviate problems due to a missed network segment. For example, entry

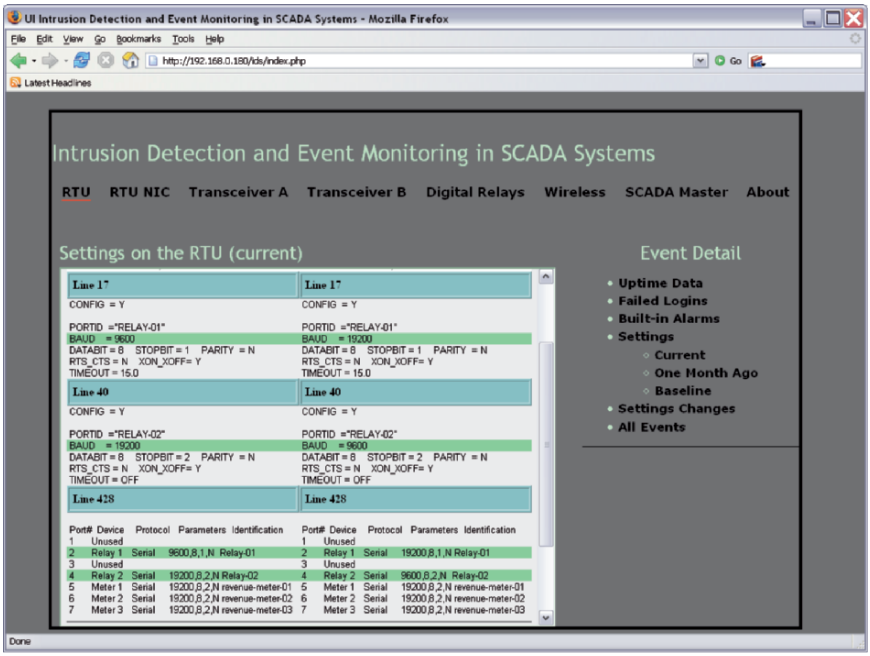

Figure 5. Screenshot of changed RTU settings.

and exit points to a network should be limited to simplify network management and reduce exposure. However, real world scenarios infrequently lend themselves to the elegant designs sought by IT professionals. Nevertheless, it is easy enough to add additional IDS sensors to each network segment. Note, however, that every additional machine, especially one providing security services, will require additional maintenance.

4.3 Revision Control

Retrieving device settings daily (or less frequently, if desired) helps archive settings for later review. Also, it enables device settings to be compared over time. This is an excellent way to guard against operator error, which is the cause of many expensive incidents in industrial environments. The security of the system is also enhanced because it is possible to determine if the settings have been changed by unauthorized parties. Most SCADA systems either do not provide this functionality or it is too difficult to implement because of the limited capabilities of SCADA devices.

Figure 5 shows a screenshot of the RTU's settings, where the baud rates for Port 2 and Port 4 have been interchanged. The two ports are directly connected to digital relays. Consequently, swapping the settings would immediately disable all communications to the relays, a very serious condition in a power substation. Subtle changes like this are often the most difficult to

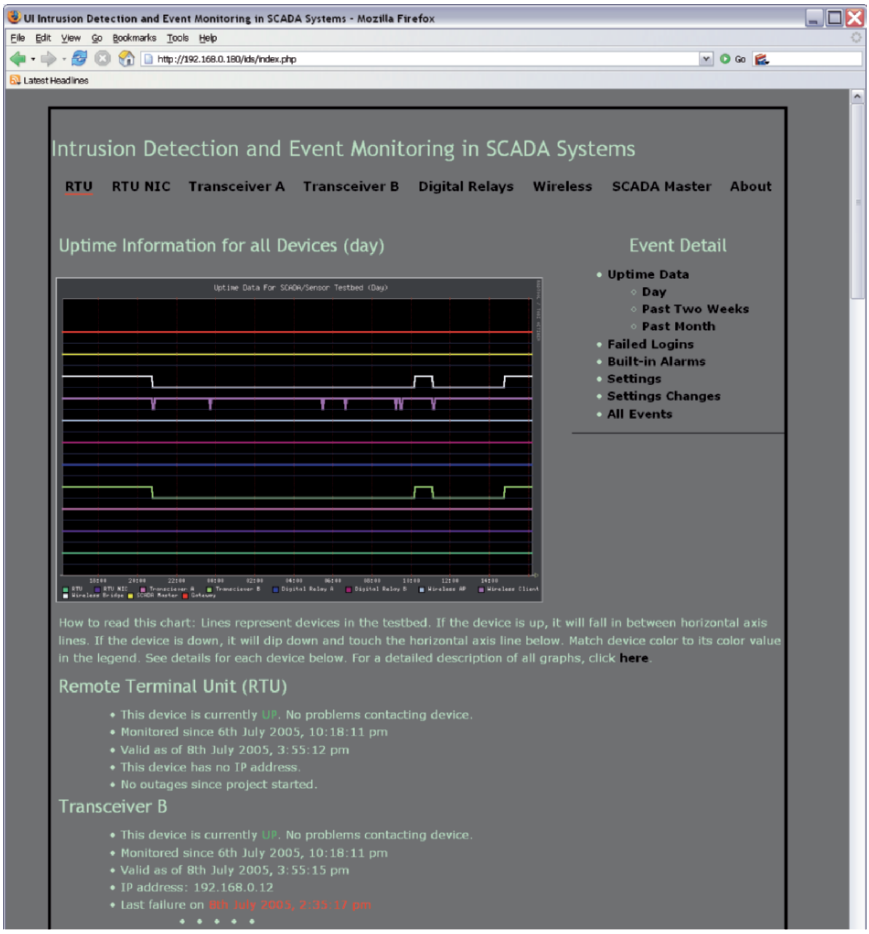

Figure 6. Screenshot of device uptimes over a 24-hour period.

detect, especially when there are dozens of relays and other devices in a network. When troubleshooting such problems, engineers usually rely on handwritten notes that may or may not be accurate. It is sometimes the case that this information was provided by another individual (or contractor) who no longer works at the facility.

4.4 Uptime Monitoring

As with most IT networks, connectivity to all SCADA network devices is essential to knowing that the communications system is healthy. Our solution provides day-, fortnight- and month-long intervals of uptime data for each device (Figure 6). Not all devices have IP addresses, so pinging some devices is not an option. However, using a Perl/Expect script, it is possible to log onto

these devices and issue a simple command – if the command succeeds, there is verification that the path is healthy.

The script for polling devices runs every five minutes and the data gathered is stored in a database. A second script graphs the data. Such graphing provides an immediate indication when a device is unreachable; even for devices that are reachable only through others. Figure 6, for example, shows that the wireless bridge is probably down, which results in transceiver B becoming unreachable. Mean time to repair (MTTR) can be calculated based on how long it takes on the average to re-establish contact.

5. Conclusions

The intrusion detection and event monitoring system is useful for securing SCADA networks as well as assisting operators in identifying erroneous or malicious settings on SCADA devices. The automated gathering and comparison of device settings over time is very useful to SCADA operators, who typically rely on personal notes and reminders about device settings.

The current prototype automates intrusion detection and settings retrieval only for RTUs. It is currently being extended to provide this functionality for other SCADA devices. Special attention will be paid to retrieving settings and detecting events involving digital relays, which are the backbone of many critical infrastructures. Our longer term goals are to place all SCADA device settings under revision control and to generate signatures for unauthorized access to other devices. Once this is accomplished, the system will be adapted to vendor-specific needs and other SCADA configurations.

Acknowledgements

This research was partially supported by the National Science Foundation under Grant DUE–0114016 (Cyber Service Training and Education) and Grant DUE–0416757 (Capacity Building for the University of Idaho Scholarship for Service Program).

References

[1] Office of Energy Assurance, 21 Steps to Improve Cyber Security of SCADA Networks, U.S. Department of Energy, Washington, DC, 2002.

[2] P. Oman, A. Krings, D. Conte de Leon and J. Alves-Foss, Analyzing the security and survivability of real-time control systems, *Proceedings of the Fifth Annual IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pp. 342–349, 2004.

[3] P. Oman, E. Schweitzer and D. Frincke, Concerns about intrusions into remotely accessible substation controllers and SCADA systems, *Proceedings of the Twenty-Seventh Annual Western Protective Relay Conference*, 2000.

[4] P. Oman, E. Schweitzer and J. Roberts, Protecting the grid from cyber attack – Part 1: Recognizing our vulnerabilities, *Utility Automation & Enginering T&D*, vol. 6(7), pp. 16–22, 2001.

[5] P. Oman, E. Schweitzer and J. Roberts, Protecting the grid from cyber attack – Part 2: Safeguarding IEDs, substations and SCADA systems, *Utility Automation & Enginering T&D*, vol. 7(1), pp. 25–32, 2002.

[6] M. Phillips, Event Monitoring and Intrusion Detection in SCADA Systems, M.S. Thesis, Department of Computer Science, University of Idaho, Moscow, Idaho, 2005.

[7] M. Roesch, Snort (www.snort.org).

[8] F. Sheldon, T. Potok, A. Krings and P. Oman, Critical energy infrastructure survivability: Inherent limitations, obstacles and mitigation strategies, *International Journal of Power and Energy Systems*, pp. 86–92, 2004.

[9] U.S. House of Representatives (Committee on Government Reform), Telecommunications and SCADA: Secure links or open portals to the security of our nation's critical infrastructure, Serial No. 108–196, U.S. Government Printing Office, Washington, DC, March 30, 2004.

[10] J. Waite, A Testbed for SCADA Security and Survivability Research and Instruction, M.S. Thesis, Department of Computer Science, University of Idaho, Moscow, Idaho, 2004.

[11] J. Waite, J. Oman, M. Phillips, S. Melton and V. Nair, A SCADA testbed for teaching and learning, *Proceedings of the Thirty-Sixth Annual North American Power Symposium*, pp. 447–451, 2004.